

TRƯỜNG ĐẠI HỌC SÀI GÒN  
KHOA CÔNG NGHỆ THÔNG TIN



## PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

---

Ứng dụng game

# Tic tac toe trên PYTHON

---

GVHD: Từ Lăng Phiêu  
SV: Võ Minh Tấn - 3120410467

TP. HỒ CHÍ MINH, THÁNG 5/2024

# Mục lục

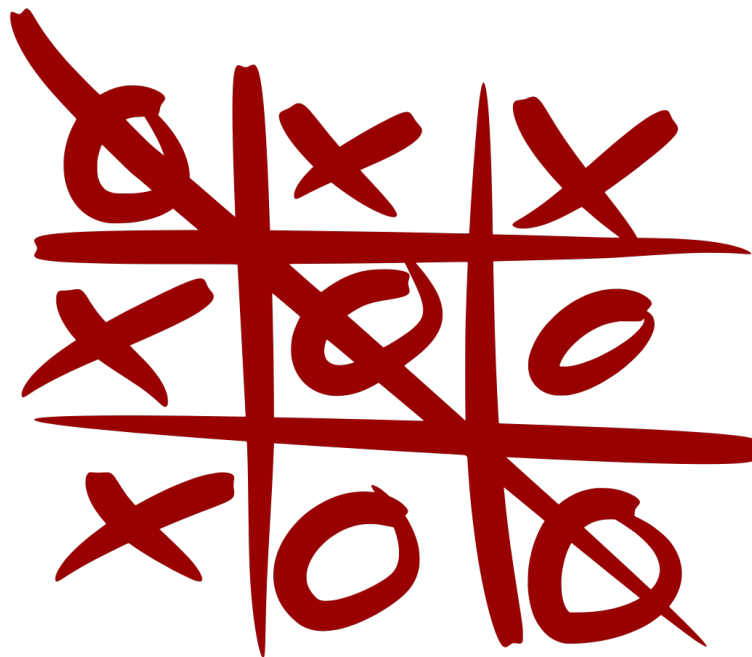
<b>1</b>	<b>Giới thiệu ứng dụng game Tic tac toe</b>	<b>2</b>
<b>2</b>	<b>Cơ sở lý thuyết</b>	<b>3</b>
2.1	Python . . . . .	3
2.2	Arcade . . . . .	4
2.3	Monte carlo tree search(MCTS) . . . . .	5
<b>3</b>	<b>Xây dựng ứng dụng game Tic tac toe</b>	<b>7</b>
3.1	Mô tả chức năng ứng dụng . . . . .	7
3.2	Cài đặt và thiết lập môi trường cho ứng dụng . . . . .	13

## 1 Giới thiệu ứng dụng game Tic tac toe

Game Tic Tac Toe, còn được gọi là X's and O's, là một trò chơi cổ điển được chơi trên một bảng vuông 3x3. Hai người chơi lần lượt đánh dấu các ô trên bảng bằng các ký hiệu X và O. Người chơi đầu tiên đạt được một hàng ngang, cột dọc hoặc đường chéo của ba ký hiệu của mình sẽ chiến thắng.

### Quy tắc trò chơi

- Trò chơi bắt đầu với một bảng vuông 3x3 trống.
- Hai người chơi lần lượt đánh dấu các ô trống trên bảng bằng ký hiệu của mình.
- Người chơi đầu tiên đạt được một hàng ngang, cột dọc hoặc đường chéo của ba ký hiệu của mình sẽ chiến thắng.
- Nếu bảng đã được đánh đầy mà không có người chiến thắng, trò chơi kết thúc với kết quả hòa.



Hình 1: Tic tac toe

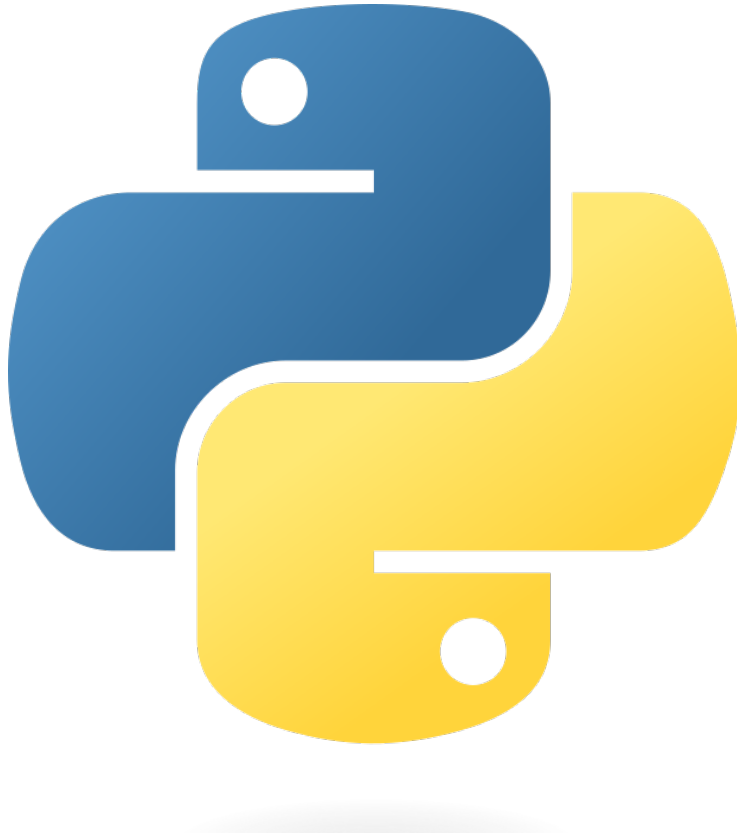
### Mục tiêu

Ứng dụng này được phát triển với mục đích nghiên cứu về hệ điều hành mã nguồn mở và cách mà socket được sử dụng trong việc thiết lập kết nối và truyền thông giữa hai máy tính. Việc sử dụng trò chơi tic tac toe, một trò chơi phổ biến và thân thuộc với nhiều người, giúp minh họa rõ hơn về cách hoạt động của socket. Điều này cũng giúp tạo ra một trải nghiệm gần gũi và thú vị cho người dùng khi tham gia vào quá trình tương tác thông qua mạng.

## 2 Cơ sở lý thuyết

### 2.1 Python

Python là một ngôn ngữ lập trình mạnh mẽ, dễ đọc và dễ hiểu được phát triển vào cuối những năm 1980 bởi Guido van Rossum. Nó được thiết kế để tăng cường tính linh hoạt và giúp các nhà lập trình tạo ra mã nguồn dễ bảo trì và tái sử dụng.



Hình 2: Logo python

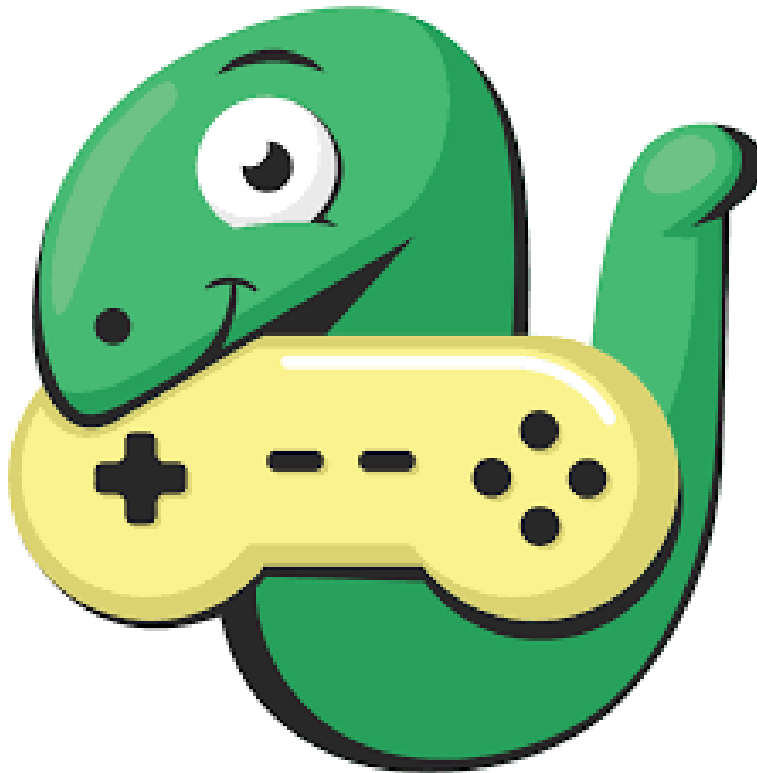
Ứng dụng của Python:

- **Phát triển Web:** Python được sử dụng rộng rãi trong phát triển web thông qua các framework như Django và Flask. Những framework này cung cấp các công cụ và thư viện hỗ trợ để xây dựng các ứng dụng web mạnh mẽ và linh hoạt.
- **Khoa học Dữ liệu và Machine Learning:** Python là một trong những ngôn ngữ phổ biến nhất trong lĩnh vực khoa học dữ liệu và machine learning. Các thư viện như NumPy, Pandas, Matplotlib, và Scikit-learn cung cấp các công cụ mạnh mẽ cho việc phân tích dữ liệu, xử lý dữ liệu và xây dựng các mô hình machine learning.
- **Ứng dụng Desktop:** Python có thể được sử dụng để phát triển ứng dụng desktop thông qua các framework như PyQt và Tkinter. Điều này cho phép nhà phát triển tạo ra các ứng dụng desktop đa nền tảng với giao diện người dùng đẹp mắt và dễ sử dụng.

- **Scripting:** Python thường được sử dụng làm ngôn ngữ kịch bản (scripting language) cho việc tự động hóa các tác vụ hoặc xử lý dữ liệu. Với cú pháp đơn giản và tính linh hoạt cao, nó là một công cụ hiệu quả cho việc viết các kịch bản tự động trong quản lý hệ thống và xử lý dữ liệu tự động.
- **Game Development:** Mặc dù không phổ biến như các ngôn ngữ khác như C++ hoặc Java trong lĩnh vực này, nhưng Python cũng được sử dụng trong phát triển game. Pygame là một thư viện Python phổ biến được sử dụng để phát triển các trò chơi đơn giản và trung bình.
- **Ứng dụng Mobile:** Python không phải là một ngôn ngữ lập trình chính thức cho việc phát triển ứng dụng di động, nhưng nó có thể được sử dụng thông qua các framework như Kivy để phát triển ứng dụng di động đa nền tảng.

## 2.2 Arcade

Python Arcade là một thư viện mã nguồn mở dành cho lập trình trò chơi và ứng dụng đa nền tảng sử dụng ngôn ngữ lập trình Python. Được phát triển bởi Paul Vincent Craven, thư viện này cung cấp một cách tiếp cận đơn giản và linh hoạt cho việc tạo ra các trò chơi 2D và các ứng dụng tương tác đồ họa. Điểm nổi bật của Arcade:



Hình 3: Logo python arcade

- **Đa nền tảng:** Python Arcade cho phép phát triển ứng dụng chạy trên nhiều nền tảng khác nhau bao gồm Windows, macOS và Linux, giúp đảm bảo sự linh hoạt và tiện lợi cho

các nhà phát triển.

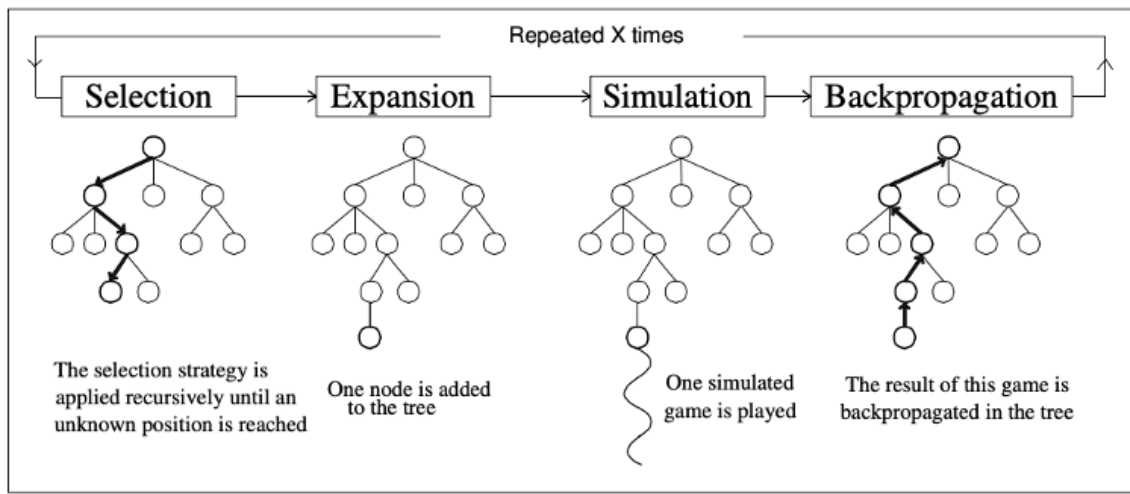
- **Hỗ trợ đồ họa mạnh mẽ:** Thư viện cung cấp một loạt các công cụ và chức năng cho việc vẽ hình, hiển thị ảnh và xử lý đồ họa 2D một cách dễ dàng và linh hoạt.
- **Xử lý sự kiện:** Python Arcade hỗ trợ xử lý các sự kiện như nhấn phím, click chuột và chạm màn hình, giúp tạo ra trải nghiệm tương tác đa dạng và phong phú cho người chơi.
- **Quản lý hiệu ứng và animation:** Thư viện cung cấp các công cụ và tính năng cho việc tạo ra các hiệu ứng đặc sắc và animation động, giúp làm cho trò chơi sống động và hấp dẫn hơn.
- **Dễ dàng sử dụng:** Python Arcade được thiết kế để dễ dàng tiếp cận và sử dụng, đặc biệt là đối với người mới bắt đầu trong lập trình trò chơi. Cú pháp và cách sử dụng đơn giản giúp giảm thời gian học và tăng hiệu quả phát triển.
- **Cộng đồng lớn:** Python Arcade có một cộng đồng người dùng và nhà phát triển lớn, với nhiều tài liệu, ví dụ và hỗ trợ trực tuyến. Điều này giúp người dùng dễ dàng tìm kiếm thông tin và giải quyết các vấn đề khi phát triển ứng dụng.

## 2.3 Monte carlo tree search(MCTS)

Thuật toán Monte Carlo Tree Search (thường được viết tắt là MCTS) là thuật toán Tìm kiếm Tốt nhất (Best-First Search) áp dụng Mô phỏng Monte Carlo. MCTS được sử dụng thay thế cho các thuật toán Cây tìm kiếm khác như Alpha-Beta khi việc tìm ra Hàm lượng giá thích hợp phải mất rất nhiều thời gian và công sức. MCTS thường được áp dụng cho các trò chơi có Thông tin hoàn hảo, gồm cả trò chơi mang tính quyết định (Go, Reversi, ...) và trò chơi mang tính ngẫu nhiên (Backgammon, các board game chiến thuật hiện đại, ...).

Nguyên lý của Mô phỏng Monte Carlo cho rằng, thông qua việc tạo dựng nên một lượng lớn các mẫu dữ liệu ngẫu nhiên và quan sát hành vi này, ta có thể ước định được hành vi của thống kê trên tập mẫu dữ liệu đó. Nói cách khác, bằng cách mô phỏng nhiều trường hợp giả định ngẫu nhiên khác nhau và lặp đi lặp lại nhiều lần, ta sẽ có được xác suất dẫn đến nhiều kết quả khác nhau. Đối với MCTS, đây là quá trình mô phỏng trận đấu với đầu vào là trạng thái trò chơi tại một thời điểm nhất định.

Kết hợp chặt chẽ với quá trình mô phỏng là Sơ đồ cây. Sơ đồ cây này lưu lại trạng thái tức thời của trận đấu cũng như nước đi cuối cùng dẫn đến trạng thái đó. Tại mỗi node chứa nước đi sẽ thực hiện mô phỏng và trả về kết quả thắng hay thua.



Hình 4: Tóm tắt các bước của MCTS

Các bước thực hiện:

- **Selection:** Chiến lược sử dụng chức năng đánh giá để chọn node tối ưu có giá trị ước tính cao nhất thông qua công thức UCT để đánh giá trong quá trình duyệt cây. Công thức đánh giá này giúp cân bằng giữa việc thăm dò và khai thác các node có tiềm năng.

Công thức UTC:

$$\frac{W_i}{n_i} + c\sqrt{\frac{\ln N_i}{n_i}}$$

- ★ **W:** Số lần thắng của nước đi
- ★ **n:** Số lần đi qua nước đi đó
- ★ **c:** Hằng số thăm dò
- ★ **N:** Số lần đi qua của node cha
- **Expansion:** Trong quá trình này, một node con mới được thêm vào cây của node được lựa chọn trong giai đoạn selection.
- **Simulation:** Trong quá trình này, từ node có được trong giai đoạn expansion, thực hiện mô phỏng ngẫu nhiên các nước đi của 2 người chơi đến khi đạt được 1 trong 3 trạng thái thắng, thua hoặc hòa.
- **Backpropagation:** Bắt đầu từ node kết quả cập nhật kết quả cho các nước đi. Thực hiện cập nhật số lần đi qua của node đó và nếu là MCTS thắng thì cập nhật số lần thắng w của cây hiện hành +1, ngược lại thua thì cập nhật số lần thắng của cây hiện hành -1. Sau khi thực hiện việc cập nhật kết quả trận đấu thực hiện đệ quy về node cha của node đó và lặp lại quá trình này cho đến khi đạt đến node gốc.

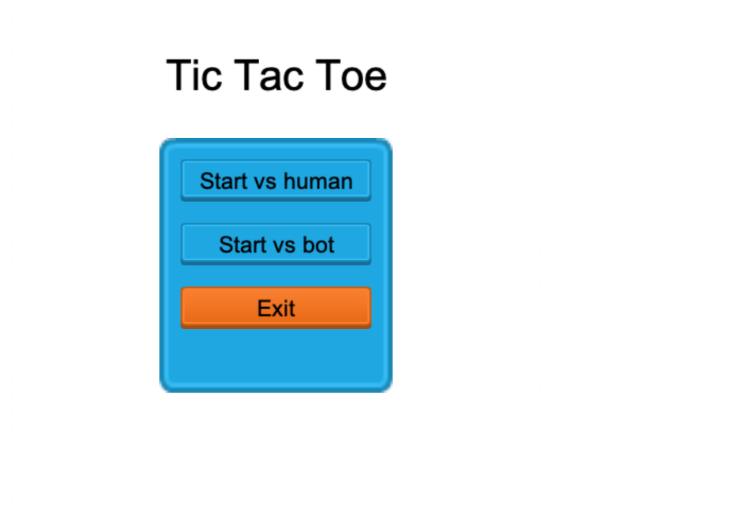
### 3 Xây dựng ứng dụng game Tic tac toe

#### 3.1 Mô tả chức năng ứng dụng

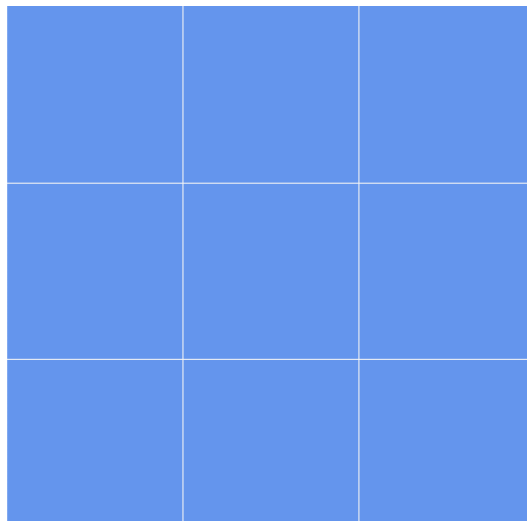
Chức năng ứng dụng:

- Đối kháng người với người (Start vs human)
- Đối kháng người với máy (Start vs bot)

Giao diện ứng dụng:

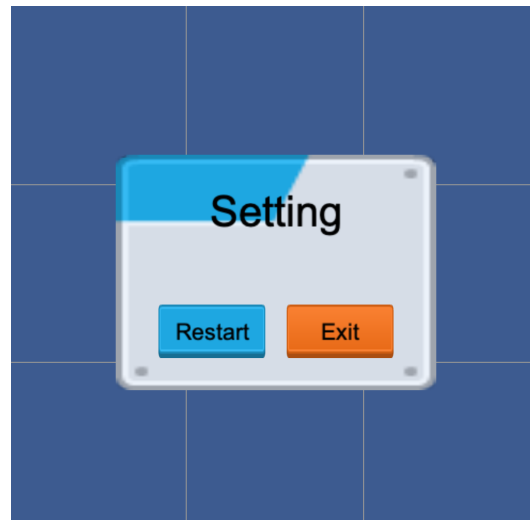


Hình 5: Giao diện chọn chế độ chơi

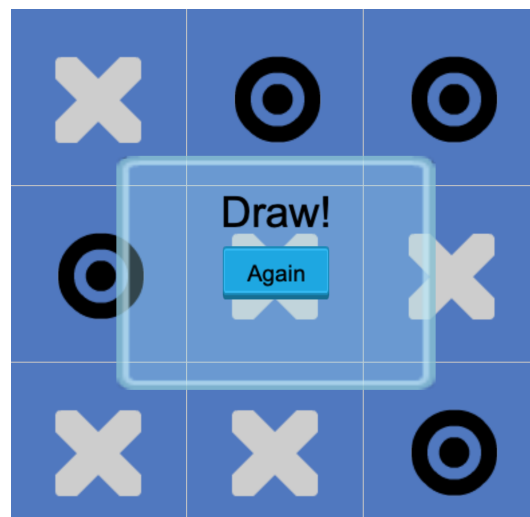


Hình 6: Giao diện board game





Hình 7: Giao diện settings



Hình 8: Giao diện end game



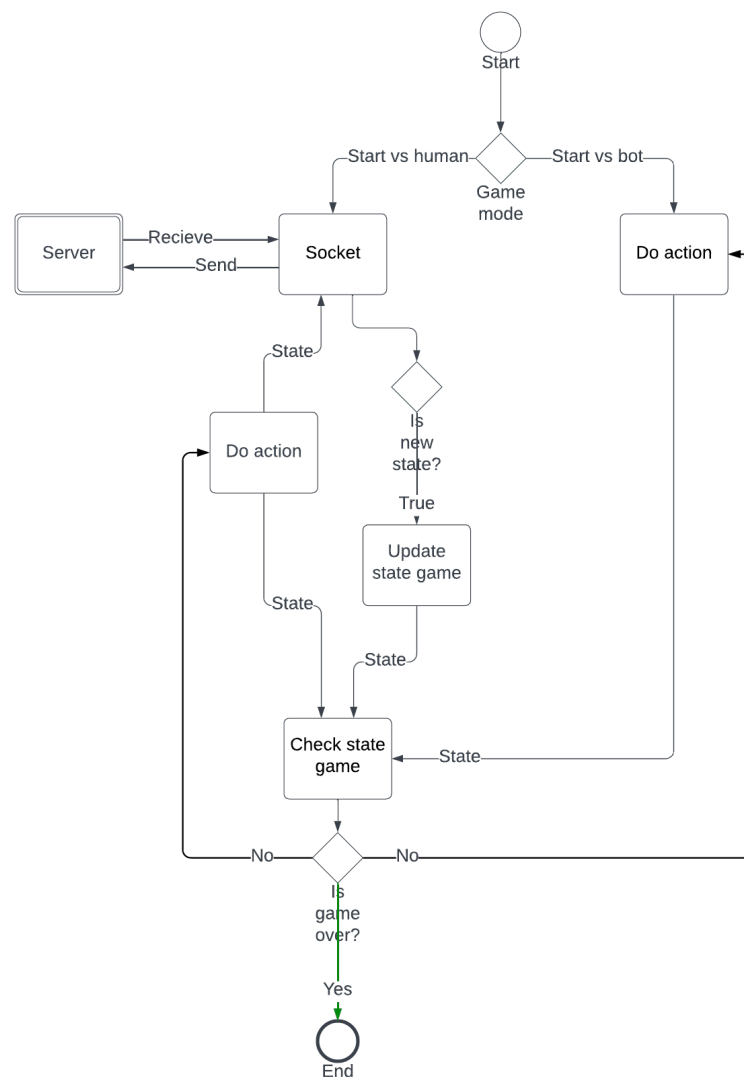
Hình 9: Cờ X



Hình 10: Cờ O

Thiết kế đơn giản với board game 3x3 không chỉ làm cho trò chơi trở nên dễ dàng để hiểu và chơi, mà còn tạo ra một trải nghiệm gần gũi và thú vị cho người tham gia. Với lưới 3x3, người chơi có thể nhanh chóng tiếp cận và tham gia vào trò chơi mà không cần quá nhiều quy tắc phức tạp. Sự đơn giản này cũng mở ra không gian cho sự sáng tạo và chiến thuật trong từng nước đi, khiến cho trò chơi trở thành một trải nghiệm thú vị không bao giờ làm chán người tham gia.

## Work flow



Hình 11: Work flow

## Đối kháng người với người (Start vs human)

Chức năng cho phép người chơi có thể chơi cùng nhau trong cùng một mạng LAN. Với kiến trúc client-server cho phép việc quản lý và bảo trì mã nguồn trở nên thuận tiện hơn bao giờ hết.

Chức năng chơi 2 người là sản phẩm của quá trình kết nối và truyền tải dữ liệu thông qua giao thức TCP, với điểm nổi bật là sử dụng socket để tạo ra ứng dụng trò chơi real time.

Dữ liệu được truyền tải qua TCP

```
1 data = {  
2     "board": self.board, % Trang thai ban co  
3     "player": self.current_player, % Nguoi cho hien tai  
4     "next_player": self.switch_player(), % Nguoi cho tiep theo  
5     "is_game_over": self.is_game_over, % Trang thai tro choi  
6     "is_new_response": True, % Kiem tra du lieu moi  
7     "is_restart": False % Trang thai restart tro choi  
8 }
```

Các dữ liệu được truyền tải từ máy khách A đến máy khách B thông qua máy chủ, được chuyển đổi thành các hành động và trạng thái của trò chơi với độ trễ cho phép tạo ra một ứng dụng trò chơi real time giữa 2 máy.

## Đối kháng người với máy (Start vs bot)

Chức năng cho phép người chơi có thể chơi với máy được xây dựng sẵn với thuật toán tìm kiếm Monte carlo tree search (MCTS). Với thuật toán này máy có thể đưa ra dự đoán và chọn nước đi phù hợp.

Lựa chọn nước đi tốt nhất

```
1 def best_action(self):  
2     time0 = time.time()  
3     while time.time() - time0 <= 2: % Lap trong 2 giay  
4         v = self._tree_policy() % Giai doan Selection va Expansion  
5         reward = v.rollout() % Giai doan Simulation  
6         v.backpropagate(reward) % Giai doan Backpropagate  
7     result = self.best_child(np.sqrt(2))  
8     return result.parent_action % Nuoc di tot nhat
```

Giai đoạn Selection và Expansion

```
1 def _tree_policy(self): % Duyet theo chieu sau va tim duong di tot nhat  
2     current_node = self  
3     while not current_node.is_terminal_node():  
4         if not current_node.is_fully_expanded():  
5             return current_node.expand() % Mo rong den khi con cua node do duoc xet
```



```
        het
6     else:
7         current_node = current_node.best_child(np.sqrt(2)) % Chọn node con tốt nhất
8     return current_node
```

---

#### Giai đoạn Simulation

```
1 def rollout(self): % Mô phỏng ngẫu nhiên các nước đi đến khi có kết quả
2     rollout_state = TicTacToe(
3         self.state.board.copy(), self.state.current_player)
4     while not rollout_state.is_game_over():
5         possible_moves = rollout_state.get_legal_actions()
6         action = self.rollout_policy(possible_moves)
7         rollout_state = rollout_state.move(
8             action, rollout_state.board.copy())
9     return rollout_state.game_result(self)
```

---

#### Giai đoạn Backpropagate

```
1 def backpropagate(self, result): % Đệ quy kết quả đến node gốc
2     self._number_of_visits += 1.
3     self._results[result] += 1.
4     if self.parent:
5         self.parent.backpropagate(-result)
```

---



## 3.2 Cài đặt và thiết lập môi trường cho ứng dụng

### Yêu cầu thiết bị

- OS: Ubuntu 22.04
- Python: Version  $\leq$  3.10.10

### Cài đặt môi trường

Tải ứng dụng:

```
$ git clone https://github.com/qwerty9927/Tic-tac-toe.git
```

Cài đặt các package cần thiết:

```
$ cd Tic-tac-toe
$ python3 -m venv .venv
$ source .venv/bin/activate
$ pip install -r requirements.txt
```

### Cấu hình server

Tắt firewall

```
$ sudo systemctl stop ufw
```

Kiểm tra IP của máy

```
$ ip a
```

```
ubuntu@ubuntu-VirtualBox:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:11:46:88 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 83660sec preferred_lft 83660sec
    inet6 fe80::cbd:2dcd:f0ec:9e16/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Hình 12: Địa chỉ IP của máy

File server.py



```
1# Config
2HOST = '10.0.2.15' % Thay doi dia chi IP cho dung voi IP cua may
3PORT = 4000
4CONNECTIONS = 2
```

File client.py

```
1# Config
2HOST = '10.0.2.15' % Thay doi dia chi IP tro den server
3PORT = 4000 % Port cua server
```



## Tài liệu

- [1] Paul Vincent Craven “**link: <https://api.arcade.academy/en/latest/>**”, *Python Arcade*, lần truy cập cuối: 30/04/2024.
- [2] Ai-boson “**link: <https://ai-boson.github.io/mcts/>**”, *Monte Carlo Tree Search (MCTS) algorithm tutorial and it's explanation with Python code*, lần truy cập cuối: 30/04/2024.
- [3] Geeksforgeeks “**link: <https://www.geeksforgeeks.org/ml-monte-carlo-tree-search-mcts/>**”, *ML / Monte Carlo Tree Search (MCTS)*, lần truy cập cuối: 30/04/2024.