tic-tac-toe Code Description

Matteo Sepa, Daniel Schiop, Lorenzo Muzaka, Samuel Hofer
Siam Islam Shomapto

Contents

1 Server

Questa pagina descrive le classi ed i metodi fondamentali del modulo Server

1.1 Classi

1.1.1 SocketConnection

La classe **SocketConnection** gestisce la connessione attraverso socket:

La classe specifica porta ed IP del server:

```
# Server IP and port
IP = "127.0.0.1"
PORT = 12345
```

Poi dichiara il tipo di socket e le opzioni di connessione:

```
# Socket type and options
server_socket = socket.socket(family=socket.AF_INET, type=socket.SOCK_STREAM)
server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

server_socket.bind((IP, PORT))
server_socket.listen()
```

1.2 Metodi

1.2.1 broadcast

Il metodo **broadcast** invia un messaggio di conferma a tutti i client connessi:

```
def broadcast(self, message, client_socket):
    """Metodo che invia un messaggio in broadcast a gli host connessi"""
    # Send messages to all clients except to the original sender
    for client in self.clients.keys():
        if client is not client_socket:
            client.send(message.encode("utf-8"))
```

1.2.2 handle

Il metodo **handle** viene chiamata per ogni client connesso e gestisce le connessioni con i client

```
def handle(self, client_socket):
    """Metodo che gestisce la connessione"""
```

1.2.3 receive

Il metodo **receive** gestisce le connessioni, i turni di gioco ed i thread per ogni client.

Il metodo inizializza la connessione, notifica il client e setta il nickname al client:

```
def receive(self):
    """Metodo che gestisce le connessioni e fa partire il thread"""
    global turn
    while True:
        # Accept Connection
        client_socket, address = self.server_socket.accept()
        print("Connected with {}".format(str(address)))
        client_socket.send(self.turn.encode('utf-8'))
        nickname = f'player {self.turn}'
```

Poi setta il turno della partita:

```
if self.turn == "x":
    self.turn = "o"
elif self.turn == "o":
    self.turn = "x"
self.clients.update({client_socket: nickname})
```

In fine avvia il thread:

```
thread = threading.Thread(target=self.handle, args=(client_socket,))
thread.start()
```