**Relazione Progetto** Traccia 3: CHATGAME Andrea Casali

0000923460

**INDICE**

1. – Introduzione
2. – Descrizione
3. – Dettagli implementativi 4 – Librerie utilizzate

# 1) INTRODUZIONE

Il progetto implementa un’architettura client-server in rete locale per il supporto di un gioco multiplayer e una grafica gui per la gestione di server e clients.

# 2) DESCRIZIONE

Sfruttando il principio della chat visto a lezione ho realizzato un gioco a domande e risposte.I giocatori sono accolti dal game\_server.py, il master del gioco, il quale richiederà l’inserimento di un nome a piacere e un indirizzo ip dato dal server stesso per collegarsi, assegnando poi un ruolo casuale ai giocatori.

Quando viene premuto il bottone “Connect” apparirà la grafica e il giocatore dovrà attendere 4 secondi prima che il timer parta e si possa iniziare a giocare.

Una volta che il timer (esso lavora in background per non disturbare il corso del gioco)inizia si dovrà scegliere una delle tre porte. Due di esse contengono una domanda mentre la terza contiene un trabocchetto che farà uscire dal gioco. Se si risponde correttamente alla domanda si guadagnerà un punto, in caso contrario si perde un punto.

Durante il gioco il punteggio sarà visibile sopra dove vengono generate le domande.

Le domande sono una serie di domande riguardanti alcuni videogiochi e alcune serie tv.

Il gioco può terminare: allo scadere del tempo, se si sceglie la porta trappola o se il giocatore decide di uscire dal gioco cliccando sul tasto “Quit”, in ogni caso viene mostarto il punteggio finale insieme a un messaggio.

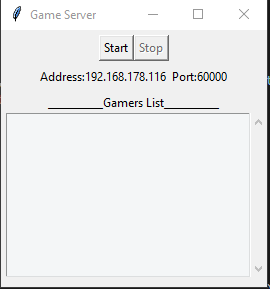
Il timer di gioco è un thread.

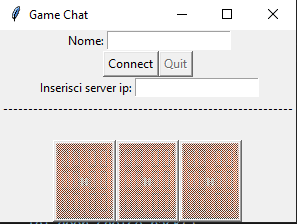
*Scelte effettuate:*

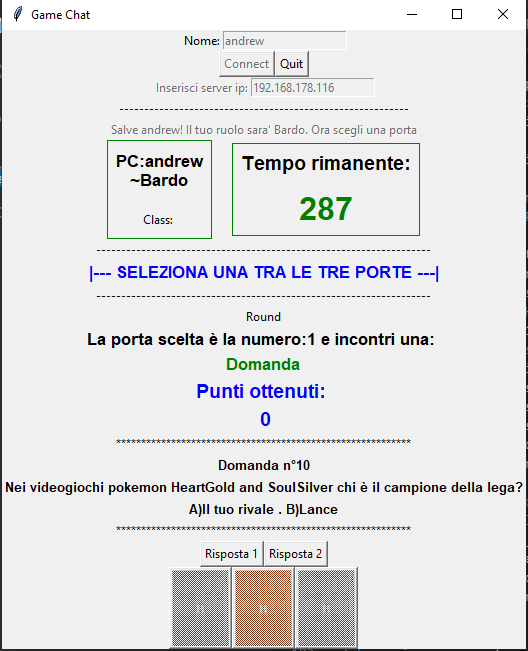
* Uso di label per dare domande e risposte con una immediata visualizzazione
* Uso di bottoni per facilitare l’inserimento dei dati da parte dell’utente e la gestione del server
* È stata scelto una durata per il gioco di 5 minuti
* Scelta della 60000 perchè vicina al limite Massimo delle porte libere quindi più probabilità di essere libera
* Uso di dizionari per facilitare la gestione dei dati

La grafica utilizzata è stata implementata usando come base il gioco RockPaperScissor del Laboratorio 12.

Di seguito alcune immagini per mostrarla:

 Grafica game\_server.py

Grafica game\_client.py prima dell’inserimento dei dati

Grafica di gioco di game\_client.py

# 3– DETTAGLI IMPLEMENTATIVI

Connessione Client-Server e porta per la socket:60000, mentre l'indirizzo IP viene inserito dall'utente, che verrà mostrato nella GUI del server.

Utilizzo le funzioni rcv() e send() della libreria socket per la gestione dell’invio di messaggi con un buffersize di lunghezza 4096 byte.

Le eccezioni vengono gestate con il sistema di try-except quali, ad esempio, l’eccezione che viene generata il timer di gioco termina oppure se il giocatore decide di cliccare il tasto “Quit ” dopo aver scelto una porta ma prima di aver scelto una risposta.

*Strutture utilizzate:*

* + liste

*Thread utilizzati:*

* + uno per il server
  + uno per client
  + uno per il timer prima dell’inizio della partita
  + un thread in background per le gestione del timer di gioco

# 4– LIBRERIE UTILIZZATE

* + Tkinter as tk,messagebox as msgbox,photoimage
  + AF\_INET, socket, SOCK\_STREAM
  + Time, sleep
  + threading
  + random