

# **Relazione Progetto**

## **Traccia 3 - Python ChatGame**

Francesco Marcaccini

0000901198

August 2021

## **Indice**

- 1. Introduzione**
- 2. Come lanciare il programma**
- 3. Modalità di comunicazione**
- 4. Modello delle Classi**
- 5. Spiegazione interfaccia grafica e funzionamento applicazione**

## **1. Introduzione:**

Questo progetto si basa sullo sviluppo di un gioco di tipo chat, nel quale i vari utenti si dovranno collegare ad un server per competere fra loro.

Il gioco è basato su 3 fasi, la prima in cui tutti i giocatori si collegano alla partita, fino a che il numero minimo di giocatori non è raggiunto.

La seconda inizia quando il numero di giocatori è stato raggiunto, a questo punto il giocatore potrà scegliere fra 3 scelte nascoste, le quali potranno rivelarsi un trabocchetto, che farà perdere il giocatore, oppure una domanda, alla quale l'utente dovrà rispondere. In caso di risposta positiva verrà aggiunto un punto, mentre al contrario in caso di risposta negativa verrà sottratto un punto.

Nella terza ed ultima fase al player verrà comunicato il proprio punteggio e in seguito verrà annunciato il vincitore.

## 2. Come lanciare il programma

Il progetto e' composto da 2 file, ossia Server.py e Client.py, i quali vanno rispettivamente lanciati tramite i seguenti comandi (utilizzando Python3):

- **Server.py:**

*"py Server.py"*

- **Client.py:**

Il client può prendere in ingresso alcuni parametri, ossia:

1. -h / -help: Per ricevere informazioni sui parametri
2. -ip / --address: Per specificare l'indirizzo IP del server  
(default="localhost")
3. -port: Per specificare l'indirizzo IP del server (default=53000)

Quindi per lanciare il client si puo' semplicemente scrivere questo comando:

*"py Client.py"*

Oppure se si volesse specificare IP o PORTA:

*"py Client.py -ip INDIRIZZO - port PORTA"*

### 3. Modalità di comunicazione

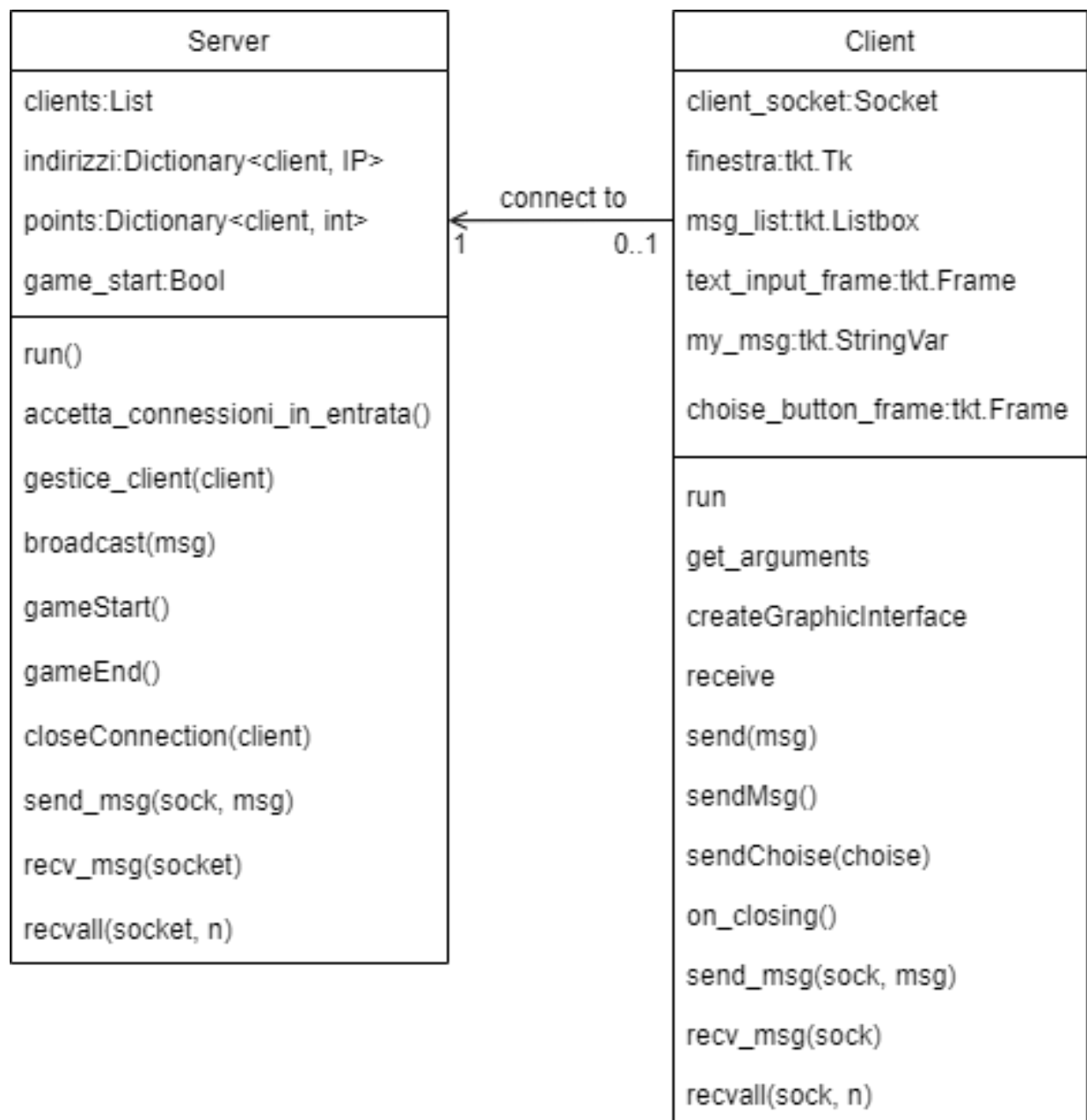
La comunicazione fra Client e Server e' gestita tramite protocollo TCP-IP e per la comunicazione dei vari messaggi ho adottato alcune regole:

1. I primi 4 byte di ogni pacchetto rappresentano la lunghezza del messaggio che e' stato mandato, così da impedire errori nella comunicazione, come l'unione di più messaggi, o la separazione di un messaggio magari troppo lungo.
2. In seguito ai primi 4 byte vi sono caratteri speciali per identificare il tipo di comunicazione, ossia per identificare a cosa il messaggio mandato sia riferito.
3. Dopo questi caratteri speciali vi e' il messaggio codificato in UTF8

*es. di pacchetto.*

Lunghezza	Tipo di messaggio	Messaggio
-----------	----------------------	-----------

## 4. Modello delle Classi



## 5. Spiegazione interfaccia grafica e funzionamento applicazione

### Lato Server:

Apri il cmd nella cartella in cui vi è il file Server.py e lancia il comando “*py Server.py*”.

```
C:\Users\User\Desktop\school\Programmazione_Di_Reti\Progetto>py Server.py  
Waiting for connections...
```

Ora il server è operativo ed in attesa della connessione dei client.

Alla connessione di un client riceveremo una notifica che l'utente “IP” si è connesso al server.

```
127.0.0.1:54828 Connected.
```

Quando tutti i giocatori necessari (in questo caso impostato a 2 per semplicità nel verificare l'applicazione) si saranno connessi riceveremo un avviso che la partita è cominciata

```
Game started.
```

Una volta finita la partita (in questo il tempo della durata della partita è settato a 30 secondi per semplicità nel verificare l'applicazione) verremo notificati che la partita è finita.

```
Game ended.
```

Alla disconnessione di un client, come per la connessione riceveremo lo stesso messaggio ma relativo alla disconnessione dell'utente.

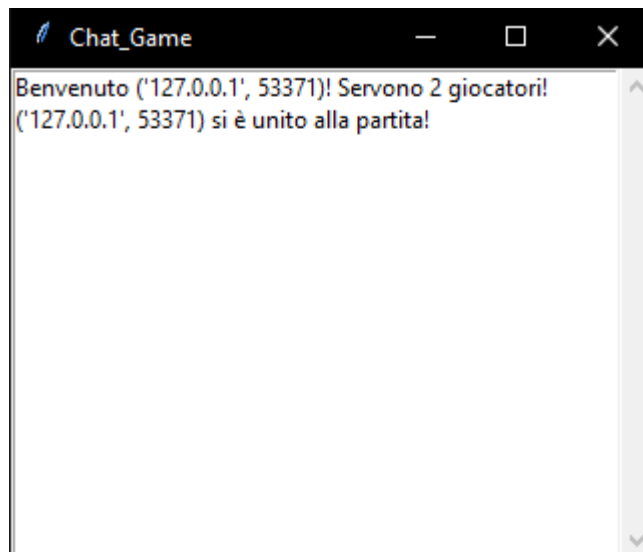
```
127.0.0.1:54828 Disconnected.
```

### Lato Client:

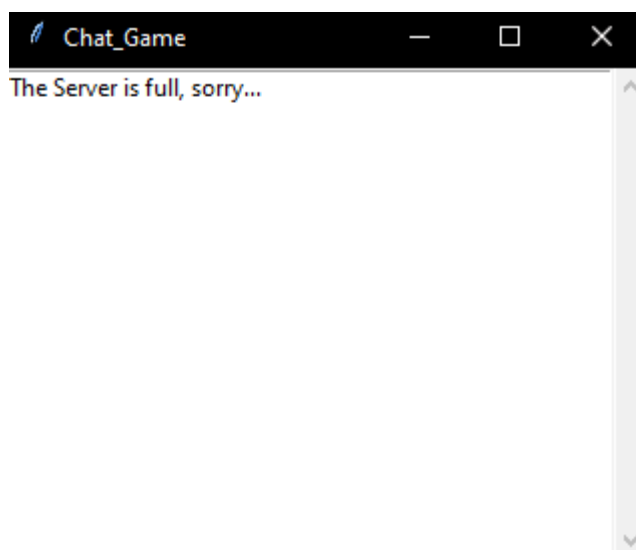
Apro il cmd nella cartella in cui vi e' il file Client.py e lancio il comando "py Client.py" con opportuni parametri se necessario.

```
C:\Users\User\Desktop\school\Programmazione_Di_Reti\Progetto>py Client.py
```

Una volta lanciato il comando si aprirà un'interfaccia grafica che permette all'utente di partecipare al gioco.

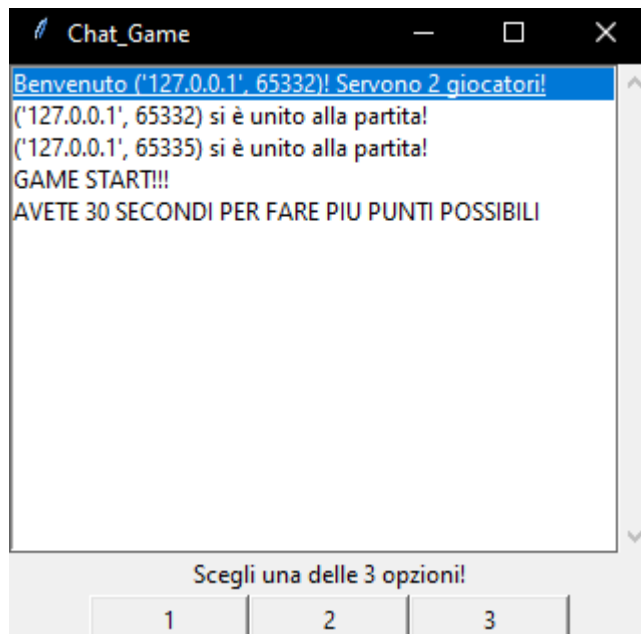


In caso il giocatore provasse a connettersi ma il server fosse già pieno o la partita fosse in corso l'utente verrebbe notificato e in seguito, dopo qualche secondo, l'applicazione si chiuderebbe.



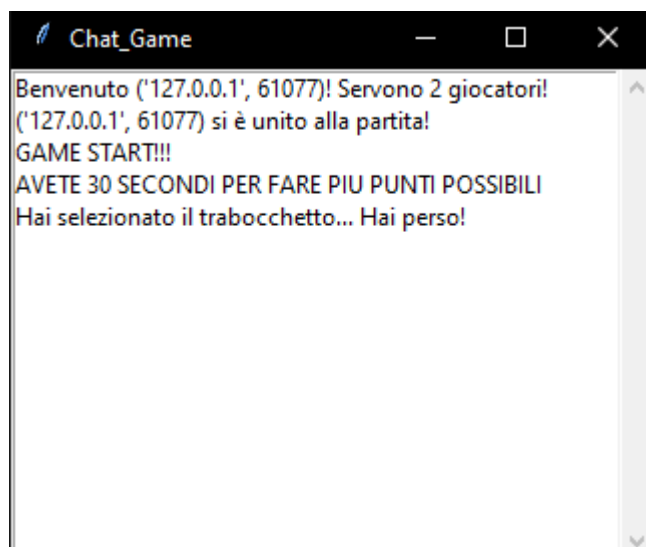


Una volta che tutti i giocatori necessari si saranno connessi la partita comincerà.

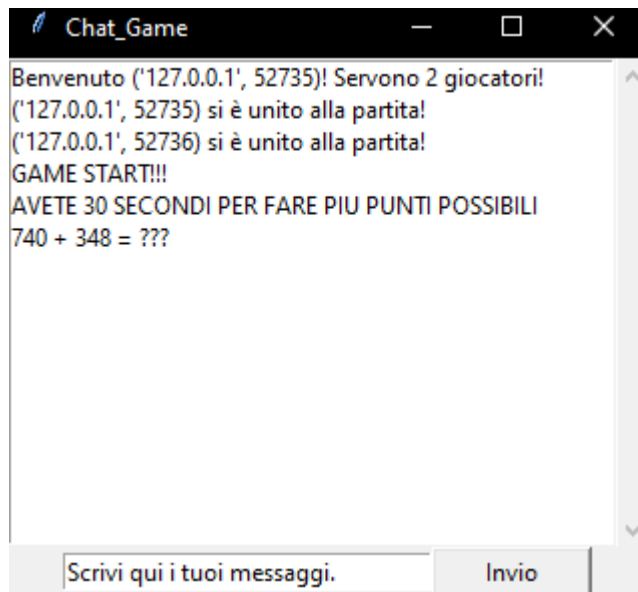


A questo punto compariranno 3 bottoni, il giocatore dovrà decidere quale premere e in caso al risultato sara' notificato se ha preso il trabocchetto, nel cui caso perderà e il gioco si chiuderà. Oppure prenderà una domanda e questa gli verrà posta.

*Caso trabocchetto*

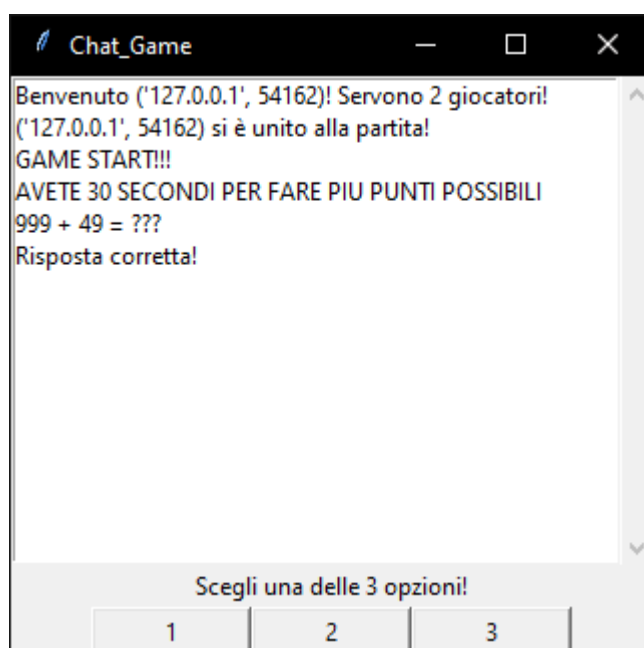


### Caso domanda

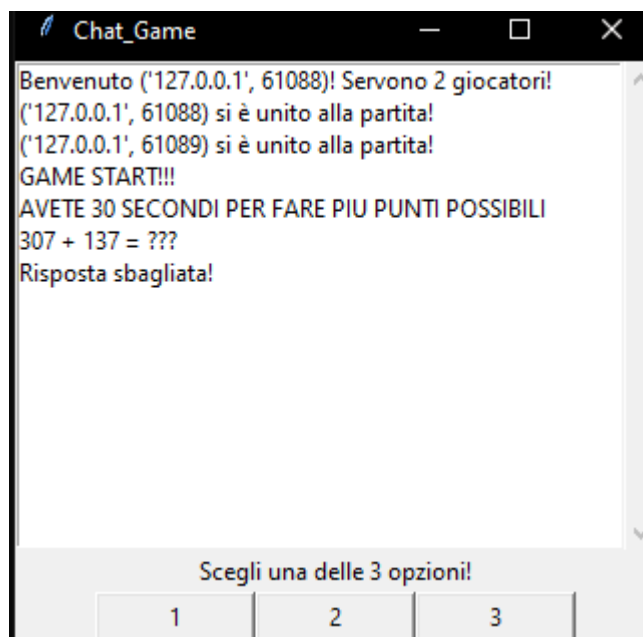


A questo punto il giocatore dovrà rispondere alla domanda che gli viene posta, se risponde in maniera esatta gli verrà notificato che ha ricevuto un punto, mentre in caso contrario gli verrà sottratto. In caso venga inserito un input diverso da quello aspettato il giocatore sarà invece disconnesso.

### Caso risposta corretta



### Caso risposta errata



Il gioco continua così sino allo scadere del tempo, dopo il quale tutti i giocatori verranno notificati con il proprio punteggio finale e il vincitore. In seguito l'applicazione si chiuderà in automatico dopo qualche secondo.

