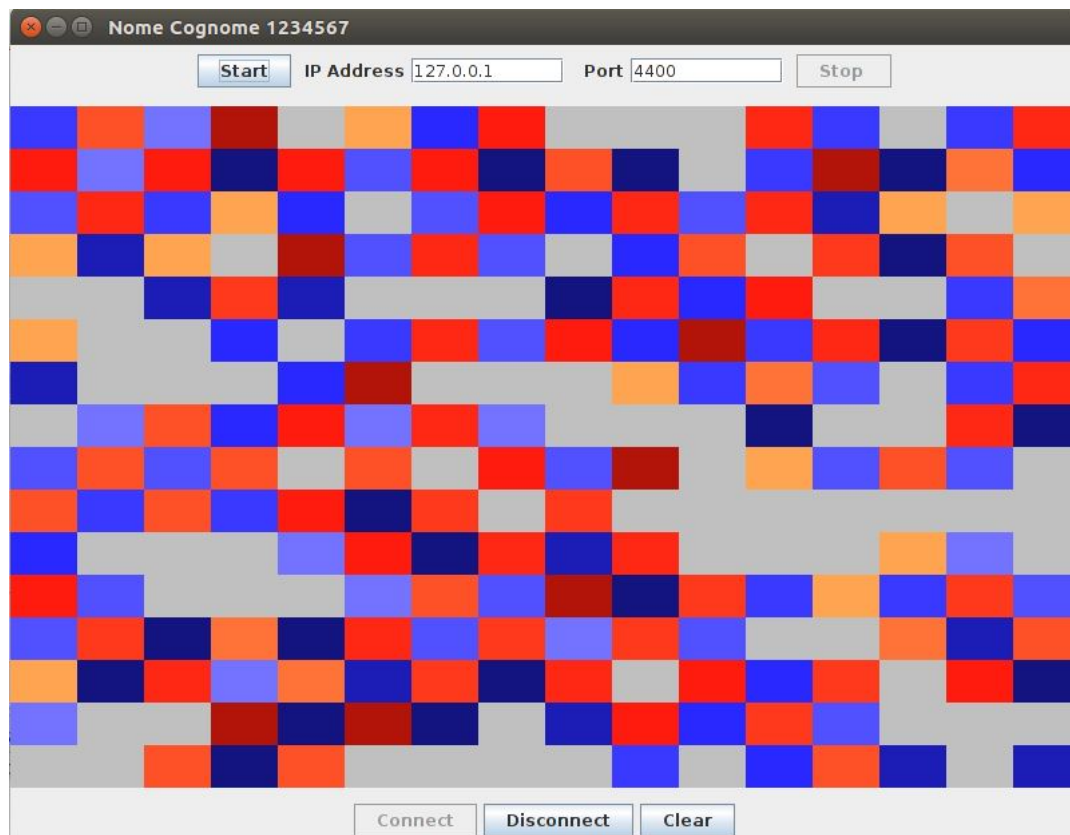


# Esercitazione

Si vuole realizzare un'applicazione client-server che permetta di visualizzare lato client un mosaico. Posizione e colore dei vari tasselli sono progressivamente comunicate dal server. Il server è multithreading ed accetta connessioni da più client. Ogni client connesso visualizza un differente mosaico (due client connessi allo stesso server di norma visualizzano differenti mosaici). I client una volta connessi possono iniziare a scaricare dal server stringhe contenenti informazioni riguardo all'indice della posizione nella scacchiera e un codice colore. Le informazioni ricevute dal server avranno impatto sul client direttamente nell'interfaccia grafica. Tutte le stringhe sono inviate da client a server e viceversa utilizzando il carattere di fine linea come separatore.

**Si richiede la realizzazione del client, con interfaccia grafica e networking.**

Il client dovrà essere composto da un frame che abbia per titolo *nome cognome e*



*matricola*  
dello  
studente e  
che  
contenga  
due campi  
testuali (IP  
Address,  
Port), cinque  
pulsanti  
(Connect,  
Start, Stop,  
Disconnect,  
Clear) ed un  
pannello che  
a sua volta  
contenga  
altri 256  
pannelli più  
piccoli (le  
tessere),  
disposti in  
una griglia  
16\*16.

All'avvio le tessere devono essere tutte del colore LIGHT\_GRAY.

Per la disposizione degli elementi si faccia riferimento all'immagine sopra.

Si implementi il seguente protocollo:

- Alla pressione del pulsante “Connect”, il client deve inviare una richiesta di connessione al server utilizzando indirizzo IP e porta indicate nei campi testuali “IP Address” e “Port”.
- Alla pressione del pulsante “Start”, dopo la connessione, il client deve mandare il comando `start` al server, il quale inizierà ad inviare ad intervalli regolari una stringa composta nel seguente modo:

*posizione\_in\_griglia;codice\_colore*

dove:

-*codice\_colore*: stringa rappresentante il codice numerico RGB del colore

-*posizione\_in\_griglia*: stringa rappresentante un intero compreso tra 0 e 255.

Se si considera la griglia come una matrice, il numero è calcolato seguendo la formula  $i*16 + j$  dove  $i, j$  sono rispettivamente gli indici di riga e colonna della matrice. Quindi ad esempio la cella [2][7] della matrice avrà *posizione\_in\_griglia* =  $2*16+7 = 39$ .

NOTA BENE: Java aggiunge gli elementi alla griglia di un pannello utilizzando l'indice calcolato seguendo il meccanismo appena descritto. Le varie componenti grafiche offrono metodi per aggiungere gli elementi in posizioni specifiche. Inoltre i riferimenti agli elementi possono essere gestiti in array (o matrici), si consiglia di utilizzare questa struttura dati per la gestione delle tessere.

Il client alla ricezione della stringa dovrà assegnare il colore corrispondente al *codice\_colore*, al pannello nella posizione indicata da *posizione\_griglia*. Per modificare il colore utilizzare il metodo `setBackground(Color c)`, per creare i colori e decodificare il *codice\_colore* fare riferimento ai metodi statici offerti dalla classe `java.awt.Color`. Consultare la documentazione per ulteriori dettagli.

Alla ricezione del *codice\_colore* = -1 e *posizione\_griglia* = -1, il client deve interrompere la ricezione dei messaggi.

Al termine dello scambio, le tessere devono rimanere colorate. In caso in cui si avvii una nuova comunicazione dopo l'interruzione (si preme di nuovo “Start”), preliminarmente il colore di ogni tessera deve essere riportato a LIGHT\_GRAY (come se si effettuasse una “Clear”, descritta in seguito).

- Alla pressione del pulsante “Stop”, dopo l'avvio, il client deve mandare al server il comando `stop`. Il server risponderà inviando una stringa che avrà *codice\_colore* = “-1” e *posizione\_griglia* = “-1”
- Alla pressione del pulsante “Disconnect” il client invia al server il comando `disconnect` e chiude la connessione
- Alla pressione del pulsante “Clear”, il colore di ogni tessera deve essere riportato a LIGHT\_GRAY

Gestire correttamente la possibilità di premere i pulsanti: in particolare

- all'avvio solo il pulsante “Connect” è abilitato
- una volta avviata una connessione, non deve poter essere premuto il pulsante “Connect”
- “Disconnect” può essere premuto solo se una connessione è avviata ma non si stanno ricevendo stringhe dal server
- “Stop” può essere premuto solo se il client è connesso e sta ricevendo stringhe
- “Start” può essere premuto solo se il client è connesso e non sta ricevendo stringhe
- “Clear” può essere premuto solo se non si stanno ricevendo stringhe dal server