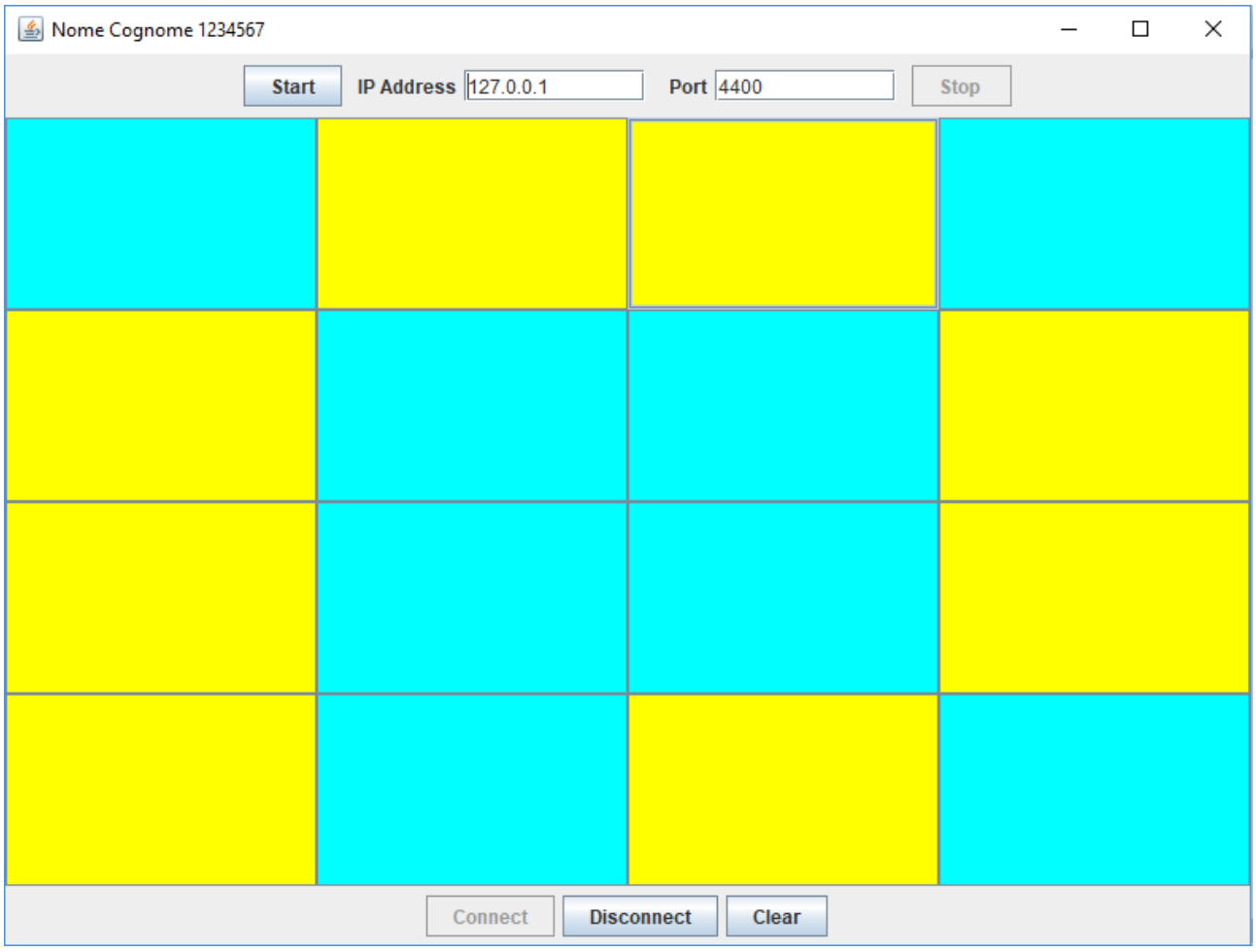
Prova Pratica

Progettazione del Software

12/09/2019 – tempo a disposizione 2h

Si vuole realizzare un'applicazione client-server che permetta di giocare ad una variante del gioco "Forza 4", il cui scopo è allineare in fila orizzontale, verticale o obliqua quattro pedine dello stesso colore. Lato client, l'utente sceglie il colore delle proprie pedine, ovvero cyan o yellow. Il server invece ha il compito di generare la partita, il quale riempirà la griglia dal basso verso l'alto, scegliendo casualmente le caselle in cui inserire le pedine di entrambi i colori. Il server è multithreading ed accetta connessioni da più clients. Ogni client connesso gestisce una partita diversa (due client connessi allo stesso server di norma gestiscono partite differenti). I client una volta connessi possono iniziare a scaricare dal server stringhe contenenti informazioni riguardo la posizione della casella nella scacchiera in cui inserire la pedina, ed il colore. Le informazioni ricevute dal server avranno impatto sul client direttamente nell'interfaccia grafica. La comunicazione è basata unicamente su scambio di stringhe. 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, interfaccia grafica e networking, in grado di comunicare con un server multithreading (fornito).

L'interfaccia grafica del client (si consiglia 800x600 come risoluzione) dovrà essere composta da un frame che abbia per titolo *nome cognome* e *matricola* dello studente, da due campi testuali (l'indirizzo IP e la porta a cui connettersi che verranno comunicati in aula), da cinque pulsanti (*Connect*, *Disconnect*, *Start*, *Stop* e *Clear*) che permettono rispettivamente di gestire la connessione col server, di disconnettersi dal server, di avviare/interrompere una sessione di gioco e di ripristinare il colore delle 16 caselle. Inoltre, l'interfaccia grafica dovrà essere composta da 1 pannello che a sua volta contiene 16 istanze di *PedinaButton* (caselle) disposte in una griglia 4*4 come in figura. Si consiglia l'utilizzo di tale classe per rappresentare le pedine sulla griglia. Lo studente, tuttavia, è libero di ignorare o modificare la classe a proprio piacimento qualora lo ritenga opportuno. All'avvio le caselle devono essere tutte del colore LIGHT_GRAY.

Gestire correttamente la possibilità di premere i pulsanti. In particolare:

- All'avvio solamente i pulsanti *Connect* e *Clear* sono abilitati. Non deve essere possibile avviare 2 connessioni dallo stesso client spingendo ripetutamente il bottone *Connect*.
- *Start* e *Disconnect* possono essere premuti solo se la connessione col server è già avviata.
- Dopo la pressione di *Start* l'unico pulsante che può essere premuto è *Stop*.
- Alla pressione di *Stop* dovranno tornare attivi solo i pulsanti *Start*, *Disconnect* e *Clear*, per effettuare una nuova partita, disconnettersi dal server, e ripristinare le 16 istanze di *PedinaButton*.
- *Clear* può essere premuto solo se non si stanno ricevendo stringhe dal server.

Si implementi il seguente protocollo:

- Alla pressione del pulsante *Connect*, il client invierà una richiesta di connessione al server utilizzando Indirizzo IP e porta indicati negli appositi campi.
- Alla pressione del pulsante *Start*, dopo la connessione, il client deve acquisire in input attraverso un messaggio in popup, il colore delle pedine scelte dall'utente (cyan o yellow). Una volta che l'utente ha inserito il colore delle proprie pedine, il client deve mandare il comando "*start*" al server, abilitare il pulsante *Stop*, e disabilitare *Start*, *Disconnect* e *Clear*. Il server inizierà ad inviare ad intervalli regolari stringhe nel seguente formato: "*posizione_in_griglia*;*colore_pedina*".
  Esempio: "3;cyan" e "7;yellow". La stringa indicante la *posizione_in_griglia*, rappresenta un intero compreso tra 0 e 15, invece *colore_pedina* il colore della pedina da visualizzare sulla griglia (i valori possibili sono due: "cyan" o "yellow"). Se si considera la griglia come una matrice, il numero rappresentante la posizione in griglia è calcolata seguendo la formula *i*4+ j* dove *i*,*j* sono rispettivamente gli indici di riga e colonna della matrice. Quindi ad esempio la cella [3][3] della matrice avrà *posizione_in_griglia* = 3*4+3 = 15.
  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. I riferimenti agli elementi, inoltre, possono essere gestiti in arrays (o matrici), si consiglia di utilizzare questa struttura dati per la gestione degli elementi in griglia.

  Il client alla ricezione della stringa dovrà cambiare colore alla casella nel pannello alla posizione indicata da *posizione_in_griglia*. Il colore diventerà Color.YELLOW se *colore_pedina* = "yellow" o Color.CYAN se *colore_pedina* = "cyan".

(**suggerimento**: Per modificare il colore si suggerisce di utilizzare il metodo *public void changeColor(Color c)* della classe *PedinaButton*).

La partita termina quando il server invia la stringa "*", ovvero quando il client avrà completamente popolato la griglia di pedine. Alla ricezione di "*", il client deve interrompere la ricezione delle stringhe, e comunicare, tramite un messaggio in popup, l'esito della partita:

1. *Vittoria utente*: se quattro pedine del colore selezionato ad inizio partita dall'utente sono allineate in fila orizzontale, verticale o obliqua;
2. *Sconfitta utente*: se quattro pedine del colore avversario sono allineate in fila orizzontale verticale o obliqua;
3. *Pareggio*: se quattro pedine sia dell'utente che dell'avversario sono allineate in fila orizzontale, verticale o obliqua. Oppure se nessuno dei due ha quattro pedine allineate.

(**suggerimento 2**: Fare riferimento al metodo public *public boolean checkColor(Color c)* della classe *PedinaButton* per verificare se la casella su cui viene richiamato tale metodo contiene una pedina di colore *c*, così da determinare l'esito della partita).

Al termine dello scambio, le caselle devono rimanere colorate. Il client dovrà inoltre disabilitare il pulsante *Stop* e riabilitare *Start*, *Disconnect* e *Clear*. Nel caso in cui si avvii una nuova comunicazione dopo l'interruzione (si prema di nuovo *Start*), preliminarmente il colore di ogni casella deve essere riportato a Color.LIGHT_GRAY (come se si effettuasse una *Clear*, descritta in seguito) ed il messaggio in popup, che chiede all'utente di inserire il colore delle proprie pedine, visualizzato.

- Alla pressione del pulsante *Stop*, dopo l'avvio, il client deve mandare al server il comando "*stop*". Il server risponderà inviando la stringa "-1". Alla ricezione di questa, il client dovrà smettere di ricevere stringhe, e comunicare, tramite un messaggio in popup, la sconfitta del client. L'interruzione della partita determina la sconfitta automatica del client. Il client dovrà inoltre disabilitare il pulsante *Stop* e riabilitare *Start*, *Disconnect*, e *Clear*.
- Alla pressione del pulsante *Disconnect*, deve essere inviata la stringa *"disconnect"*, e devono essere chiusi i canali di comunicazione generati in fase di connessione. Deve inoltre essere abilitato nuovamente il pulsante *Connect* in quanto deve essere possibile instaurare una nuova connessione senza che sia necessario il riavvio del client.
- Alla pressione del pulsante *Clear*, il colore di ogni casella deve essere riportato a Color.LIGHT_GRAY