

Progettazione del Software

Prova Pratica

10/09/2020 - tempo a disposizione 2h

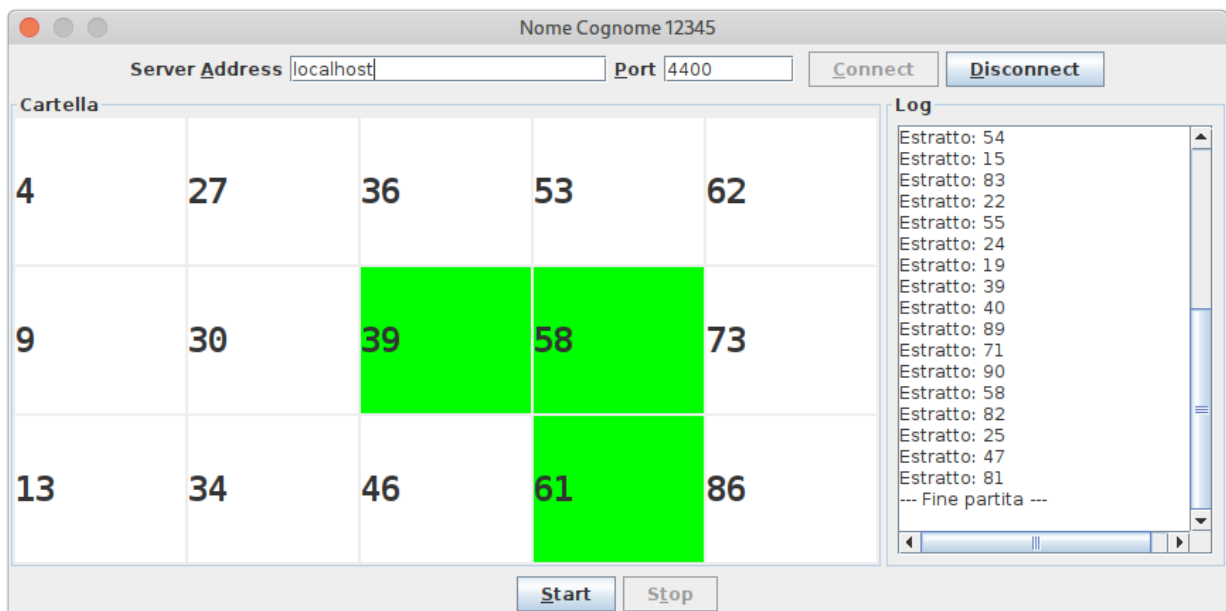
Si vuole realizzare un'applicazione *client-server* che permetta di giocare ad una variante semplificata del gioco del *Bingo* (o Tombola).

Il gioco si svolge nel seguente modo: 1) Il giocatore (rappresentato dal client) genera casualmente una sequenza di 15 numeri da 1 a 90 senza ripetizioni, disposti secondo una matrice 3×5 che formano la cosiddetta *cartella* di gioco. 2) Il banco (rappresentato dal server) si occupa solamente dell'estrazione dei numeri. Per ciascuna partita il banco estrae, in modo casuale, 30 numeri compresi tra 1 e 90, senza ripetizioni e li invia al client (si immagini l'estrazione di 30 palline da un sacchetto di 90 palline, numerate da 1 a 90). 3) Ogni qualvolta il banco estrae un numero presente nella cartella del giocatore, il giocatore lo segna, cambiando il colore della cella corrispondente.

Il server è *multithreading* ed accetta connessioni da parte di più client, e ciascun client connesso gestisce una partita diversa. 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.

Durante l'esame il server sarà raggiungibile al seguente indirizzo:

- Indirizzo IP: 80.211.232.219
- Porta: 4400



Si richiede la realizzazione del client, con interfaccia grafica e networking, in grado di comunicare con il server multithreading (fornito).

L'interfaccia grafica del client dovrà essere composta da un frame che abbia come titolo *nome cognome matricola* dello studente, da un pannello centrale contenente la cartella di gioco, da un pannello laterale contenente un'area di testo che rappresenti il log di gioco, da due campi testuali editabili per l'indirizzo, la porta del server, e da quattro pulsanti che realizzino le seguenti funzioni: 1) *Connect*: permette di connettersi al server utilizzando indirizzo e porta specificati nei campi testuali; 2) *Disconnect*: permette di chiudere la connessione al server; 3) *Start*: permette di avviare una nuova partita; 4) *Stop*: permette di interrompere la partita corrente.

Il pannello centrale rappresenta la cartella di gioco ed è composto da 15 istanze della classe fornita `TicketCell`, disposte secondo una griglia 3×5 . La classe `TicketCell` è una specializzazione di `JPanel` che rappresenta la singola casella della cartella di gioco. Ciascuna istanza di `TicketCell` contiene un `JLabel`, ed un intero rappresentante il suo numero (accessibile tramite i metodi `getValue()` e `setValue(int)`), che viene visualizzato tramite il `JLabel` nell'interfaccia grafica, mentre la selezione o meno della casella è realizzata cambiando il colore di background del pannello (`Color.WHITE` non selezionato, `Color.GREEN` selezionato). Il pannello di log è composto da un'istanza di `JTextArea`, non editabile, che mostra il log degli eventi del gioco (numeri estratti, partita iniziata, ecc...). Si consiglia di impostare la textarea con 20 righe, 30 colonne.

Suggerimento: Per far sì che sia possibile visualizzare più di 20 righe del log, si suggerisce di abilitare lo *scrolling* per la textarea, incapsulandola all'interno un'istanza di `JScrollPane`. (si faccia riferimento alle API di Swing per `JScrollPane`).

Suggerimento: Per ottenere il corretto posizionamento si utilizzi `BorderLayout` per il frame del client, aggiungendo il pannello centrale in posizione `BorderLayout.CENTER`, ed il pannello di log in posizione `BorderLayout.EAST`.

Si implementi il seguente protocollo:

- All'avvio solamente il pulsante *Connect* deve essere abilitato.
- Alla pressione del pulsante *Connect*, il client invia una richiesta di connessione al server utilizzando *indirizzo e porta indicati negli appositi campi*.
- Una volta stabilita correttamente la connessione, il client deve abilitare i pulsanti *Start* e *Disconnect*. Alla pressione del pulsante *Start*, il client deve inviare la stringa "start" al server, abilitare il pulsante *Stop*, e disabilitare i pulsanti *Start* e *Disconnect*. Il client dovrà inoltre generare una nuova cartella, assegnando a ciascuna cella un numero casuale tra 1 e 90, senza ripetizioni, e deselezionare ogni eventuale cella selezionata in una partita precedente (si faccia riferimento al metodo `TicketCell.setSelected(boolean)` nella classe fornita). Il popolamento delle celle deve avvenire in modo tale che i numeri siano disposti in ordine crescente, dall'alto verso il basso, e da sinistra verso destra (vedi figura). Il contenuto della textarea di log non deve invece essere cancellato.

Suggerimento: Per ottenere l'ordinamento dei numeri nella cartella è sufficiente ordinare la sequenza di numeri casuali, ed inserirli scorrendo le celle prima per colonna, poi per riga.

Esempio: Data una sequenza di numeri 10 numeri casuali generati:

36, 14, 22, 66, 31, 12, 44, 85, 20, 55

Una volta ordinata la sequenza è sufficiente popolare la matrice procedendo prima per colonna, e poi per riga:

12, 14, 20, 22, 31, 36, 44, 55, 66, 85

12	36
14	44
20	55
22	66
31	85

- Una volta ricevuta la stringa “start”, il server inizierà ad estrarre 30 numeri da 1 a 90 senza ripetizioni e ad inviarli al client, ad intervalli regolari, uno alla volta.
Alla ricezione di ciascun numero, il client deve verificare se tale numero è presente nella propria cartella, ed in caso affermativo, evidenziare la cella corrispondente cambiandone il colore di background a `Color.GREEN` (si faccia riferimento al metodo `TicketCell.setSelected(boolean)` nella classe fornita). Il client dovrà inoltre aggiungere alla textarea di log la stringa “Estratto: num” dove num è il numero restituito dal server. **Suggerimento:** Al fine di semplificare la fase di debug si suggerisce di far stampare a terminale tutte le stringhe ricevute dal server.
- Alla pressione del pulsante *Stop*, durante una partita, il client dovrà inviare la stringa “stop” al server. Alla ricezione della stringa “stop”, il server provvederà ad interrompere immediatamente la partita corrente, inviando la stringa “+” al client, ad indicare l’avvenuta terminazione. Qualora invece il server termini di inviare i 30 numeri al client allo stesso modo invia la stringa “+”, ad indicare la corretta terminazione.
- Alla ricezione della stringa “+”, il client deve interrompere la ricezione dal server, abilitare i pulsanti *Start* e *Disconnect*, ed aggiungere la stringa “Fine partita” al contenuto della textarea di log.
- Alla pressione del pulsante *Disconnect*, il client deve inviare la stringa “disconnect”, chiudere tutti i canali di comunicazione generati in fase di connessione, e deve inoltre abilitare nuovamente il pulsante *Connect* in quanto deve essere possibile instaurare una nuova connessione senza che sia necessario il riavvio del client.