TECNICHE DI PROGRAMMAZIONE

05/03/19

Programma “Indovina il numero”.

Per prima cosa bisogna ragionare sugli elementi che potrebbero servirci nel nostro programma. Sicuramente abbiamo bisogno:

* Di un bottone ‘Nuova partita’ per iniziare una nuova partita.
* Fornire all’utente un modo per scrivere un tentativo.
* Modo per visualizzare i tentativi rimanenti.
* Modo per indicare all’utente che il tentativo era TROPPO ALTO o TROPPO BASSO.

Quante azioni può fare l’utente? Due: iniziare una nuova partita e inserire un tentativo.

Creiamo il nuovo progetto JavaFx ( File-New…-Other…-JavfaFx Project).

Ricordarsi nell’ ultima schermata della creazione del Java Project:

- Language 🡪 FXML

- Package Name 🡪 it.polito.tdp.(nome)

Una volta creato il progetto, per prima cosa lavoro sul file FXML ( lo apriamo sempre con SceneBuilder ).

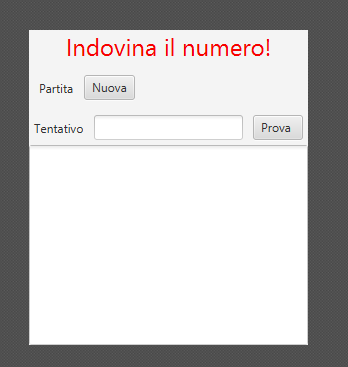
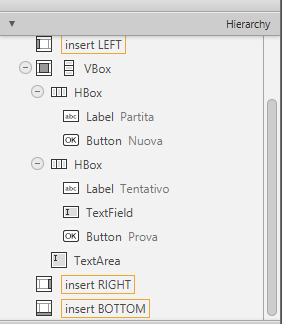
Per prima cosa aggiungiamo una label nel BorderPane per il titolo ( Indovina il numero! ).

Ci serviranno inoltre:

* Un bottone ‘inizia partita’.
* Una casella di testo con un suo bottone per inserire il tentativo.
* Un’ area di testo per visualizzare i messaggi.
* Qualcosa per visualizzare il numero di tentativi rimasti.

I passaggi sono i seguenti:

1. Inseriamo un Vbox nella parte centrale del BorderPane in quanto sappiamo già che la nostra finestra avrà una struttura che si sviluppa verticalmente.
2. Inseriamo 2 Hbox all’interno del Vbox in modo da dividere quest’ultimo in 2 parti; una parte superiore in cui metto i controlli per gestire la partita e una inferiore per gestire i tentativi.
3. Inseriamo un area di testo ( TextArea ) sempre all’interno del Vbox sotto gli Hbox per la visualizzazione dei messaggi.
4. Infine aggiungiamo tutti gli altri elementi in modo da ottenere il seguente schema.

Ricordarsi di separare bene gli elementi tra di loro usando il Layout sulla destra e di adeguare le dimensioni dei Vbox e degli Hbox (USE\_COMPUTED\_SIZE ).

Aggiungiamo ancora nell’ Hbox piu il alto un’altra label e un altro TextField per indicare i tentativi rimasti.

Questa ultima TextField sui tentativi rimasti non deve essere modificabile dall’utente quindi togliamo la spunta nel comando Editable delle proprietà. Lo stesso vale per la TextArea.

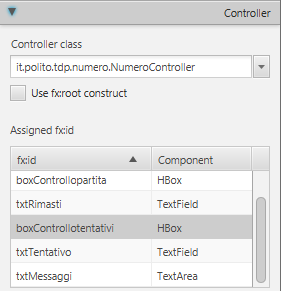
Una cosa molto importante che dobbiamo essere capaci di controllare è quella di evitare all’utente di premere il bottone ‘Prova’ se non è stato prima premuto il tasto ‘Nuova’. Per fare questo disabilitiamo il bottone prova ( meglio tutta l’Hbox ), spuntando ‘ Disable ‘ nelle Proprietà.

Iniziamo quindi a dare i nomi agli elementi che ci servono nel codice Java. Ai bottoni non mi interessa dare un ID in quanto l’unica cosa che fanno è generare un’azione. Nel caso in cui dovessi abilitarli o disabilitarli, invece, sarei costretto a creare un ID in modo tale da poter fare SetDisable TRUE o SetDisable FALSE.

Nel nostro caso noi abbiamo deciso di spuntare Disable alle Hbox ( vedi due paragrafi sopra ). Dobbiamo quindi decidere un ID per le Hbox e non per i bottoni.

Selezioniamo quindi il primo Hbox e nella finestra Code in basso a destra andiamo a mettere come fx:id ‘boxControllopartita’. L’ fx:id del secondo Hbox è invece ‘boxControllotentativi’.

Aggiungiamo anche un fx:id al TextField del primo Hbox ( quello che indica i tentativi rimasti ) ‘txtRimasti’, al TextField del secondo Hbox ‘txtTentativi’ e alla TextArea ‘txtMessaggi’. Per controllare che abbiamo fatto tutto correttamente possiamo andare in Controller in basso a sinistra. Dovremmo visualizzare la seguente schermata:

Aggiungiamo un’azione On Action ‘handlePartitaNuova’ sul bottone Nuova per gestire la pressione del bottone appunto. La stessa cosa va fatta con il bottone Prova inserendo l’azione ‘handreProvaTentativo’.

Una volta controllato di aver inserito tutti i fx:id e gli handle, possiamo finalmente farci generare lo scheletro del programma da:

View-Show Sample Controller Skeleton

E infine premere su Copy per copiarlo (Ricordarsi di spuntale ‘full’ prima di copiarlo). Questo scheletro del programma lo andremo ad incollare dentro Eclipse nel file NumeroController (togliere prima le 2 righe vecchie ).

Da adesso in poi lavoreremo sul file NumeroController di Eclipse e non piu su SceneBuilder.

Innanzitutto, andiamo a definire le variabili java che ci serviranno nel programma e su cui dovremo andare a fare dei controlli. Dopo :

**public** **class** NumeroController {

inseriamo:

**private** **final** **int** NMAX =1000;

**private** **final** **int** TMAX =8;

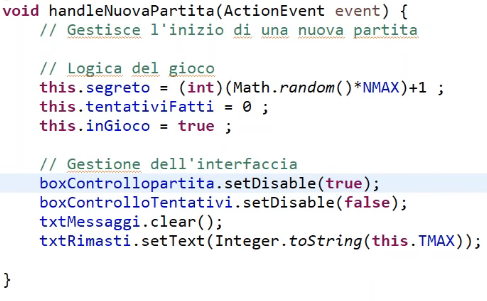
Queste due variabili Nmax e Tmax rappresentano rispettivamente il numero massimo che possiamo indovinare (il numero da indovinare deve essere compreso tra 0 e 1000) e il numero massimo di tentativi disponibili.Queste variabili non sono a controllo dell’utente, non verranno visualizzate nell’interfaccia.

Creiamo inoltre la variabile ‘segreto’ che conterrà il valore del numero segreto da indovinare e la variabile tentativiFatti che indicherà il numero di tentativi eseguiti dall’utente.

**private** **int** segreto;

Definiamo inoltre una variabile booleana ‘inGioco’ per sapere se la partita è gia iniziata oppure no.

**private** **boolean** inGioco=**false** ;

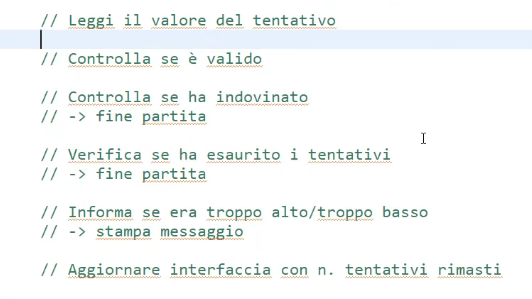


In figura: GESTIONE del Bottone ‘Nuova’, spiegazione sotto.

Spiegazione: Una volta premuto il bottone ‘Nuova’ , noi vogliamo che il programma crei un numero tra 1 e 100; quest’ultimo sarà il nostro numero segreto da indovinare. Per fare questo usiamo il comando Math.random() che crea un numero decimale tra 0 (compreso) e 1 (escluso). Moltiplicando questo valore per Nmax (100) , otteniamo un numero compreso tra 0 e 99 (DOUBLE o FLOAT) . Aggiungendo (int) prima di questo comando, noi otteniamo un numero intero tra 0 e 99. Infine aggiungiamo 1 perché il nostro numero deve essere compreso tra 1 e 100. Inoltre impostiamo il numero di tentativi fatti dall’utente = 0 e la variabile booleana = true in modo tale da indicare che l’utente è in gioco.

Per quanto riguarda l’interfaccia, disabilitiamo l’ Hbox che contiene il bottone Nuova e il numero di tentativi rimasti cosicché l’utente non possa modificarlo. Comando setDisable(true). Mediante la doppia negazione setDisable(false) abilitiamo l’Hbox che permette all’utente di inserire un tentativo. Puliamo l’area di messaggi con il comando clear. Mediante l’ultimo comando inseriamo il numero di tentativi massimi che ha l’utente per indovinare ( successivamente vedremo come decrementarlo) . Qui dobbiamo ricordarci che txtRimasti accetta solo stringhe, mentre Tmax è un int; grazie a Integer.toString possiamo convertire il nostro numero in stringa.

Per quanto riguarda l’handler ‘ProvaTentativo’, l’altro bottone, noi vogliamo che faccia le seguenti cose:

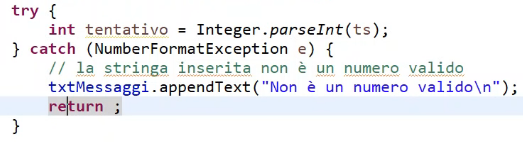
ù

Per leggere il valore del tentativo, creiamo una variabile ts che conterrà il valore inserito dal utente (STRINGA).

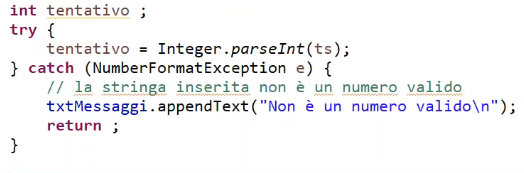
String ts = txtTentativo.getText();

Successivamente creiamo un’altra variabile di tipo integer chiamata ‘tentativo’ che conterrà la stringa contenuta in ts ma convertita in integer mediante il comando parseInt.

**int** tentativo=Integer.*parseInt*(ts);



Questa operazione di conversione la inseriamo tra un “ try-catch” per gestire eventuali errori che l’utente potrebbe commettere nell’inserimento ( per esempio sbaglia a premere sulla tastiera ). Nel caso di errore viene visualizzato nella TextArea la scritta “Non è un numero valido “.

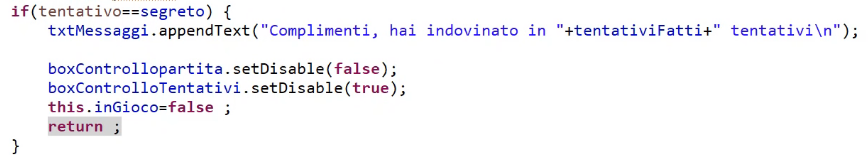


Stiamo attenti alla dichiarazione della variabile tentativo! Avessimo lasciato come prima, la variabile sarebbe stata definita solo all’interno del try-catch (non avremmo potuto usarla fuori da esso!).

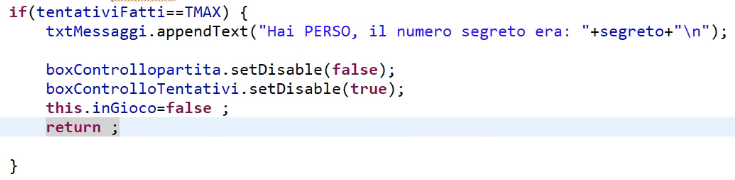
Successivamente incrementiamo la variabile tentativiFatti usando il codice:

tentativiFatti++;

Con il codice riportato in seguito, invece, controlliamo se il tentativo fatto è uguale al numero segreto e in caso positivo ci congratuliamo con l’utente ( fine del gioco ). Inoltre abilitiamo l’Hbox superiore per permettere all’utente di giocare una nuova partita e disabilitiamo quello inferiore rendendolo cosi immodificabile dal’utente ( ritorniamo alle condizioni iniziali prima di avviare una partita). Con il comando this.inGioco=false diciamo che il nostro utente non è piu in gioco.



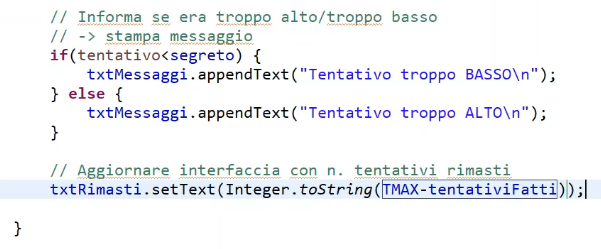
Dobbiamo adesso gestire il caso in cui l’utente abbia terminato i tentativi a sua disposizione. Per fare ciò utilizziamo il seguente codice:



Impostiamo tentativiFatti==TMAX. Se così fosse, scriviamo nella textArea il messaggio hai PERSO….

Anche in questo caso il gioco sarebbe finito e quindi inseriamo gli stessi comandi visti sopra.

Il seguente codice gestisce il caso di tentativo più alto o più basso rispetto al numero segreto (non c’è molto da dire). L’unica cosa importante è ricordarsi di trasformare la variabile TMAX-tentativiFatti da integer a String in quanto la stiamo inserendo in una textField (txtRimasti).



FINE