

AlgaT

Relazione progetto

Introduzione

L'applicazione è suddivisa in 2 parti principali che rappresentano, concettualmente, le 2 parti fondamentali del tutorial:

1. **Lezioni:** all'interno di questa sezione l'utente trova tutte le lezioni previste dal tutorial le quali sono costituite da una parte scritta che tratta l'argomento dal punto di vista meramente teorico e una parte di domande grazie alle quali lo studente può verificare il suo grado di conoscenza dell'argomento.
2. **Playground:** questa sezione, che l'utente può raggiungere in qualsiasi momento grazie ad un pratico pulsante nell'interfaccia delle lezioni, permette di simulare inserimenti e rimozioni all'interno di una Tabella Hash opportunamente configurata.

La scelta di separare completamente le due sezioni è sia didattica che implementativa. Infatti, se da un lato permette all'utente di interrompere in qualsiasi momento lo studio della teoria, in favore di un approfondimento più "concreto" dei concetti già studiati, dall'altro costituisce una notevole agevolazione a livello implementativo in quanto ha consentito di realizzare le due sezioni in modo totalmente indipendente l'una dall'altra.

Lezioni

Tra i requisiti del progetto figura la facilità di aggiunta di nuovi contenuti: la scelta è pertanto ricaduta su YAML, un formato di serializzazione dei dati estremamente human-friendly, sia sul lato lettura che sul quello di scrittura. Grazie ad una libreria Java (SnakeYAML) è stato inoltre possibile specificare, attraverso opportune classi Java, il modello dei dati atteso. In particolare si è definita una classe Lesson contenente campi come il titolo della lezione, il contenuto della stessa e la lista delle domande definite anch'esse grazie ad una classe Question con i campi relativi.

Durante l'inizializzazione AlgaT legge e interpreta quindi il file YAML e crea gli oggetti necessari a rappresentare i dati contenuti nel file all'interno dell'applicazione.

L'interfaccia (Fig. 1) mostra quindi il bottone che conduce al Playground, la lezione o le domande (che compaiono al centro della schermata) ed un pannello con i collegamenti alle rimanenti lezioni e domande.

Il contenuto della lezione propriamente detta, viene ricavata dal documento HTML lesson**Numero** e caricato in una WebView. Si è scelto di procedere in questo modo per la facilità di formattazione e utilizzo offerti dal linguaggio HTML e per la sua larghissima diffusione.

Le domande non fanno parte del file HTML e sono invece create dinamicamente in seguito al click del pulsante corrispondente. Inizialmente solo la prima domanda viene mostrata all'utente e, solo in seguito ad una risposta corretta, appare la successiva. Questa funzionalità è resa possibile dall'esecuzione di una funzione in risposta al click dell'utente su uno dei RadioButton che rappresentano le risposte alla domanda. Oltre a gestire la comparsa della domanda successiva questa funzione si occupa anche di evidenziare di verde o di rosso la risposta dell'utente a seconda che sia o meno corretta. In seguito ad una risposta corretta all'ultima domanda i pulsanti legati alla lezione successiva vengono abilitati in modo tale che l'utente possa avanzare nel tutorial solo dopo aver verificato con successo le sue conoscenze sull'argomento.

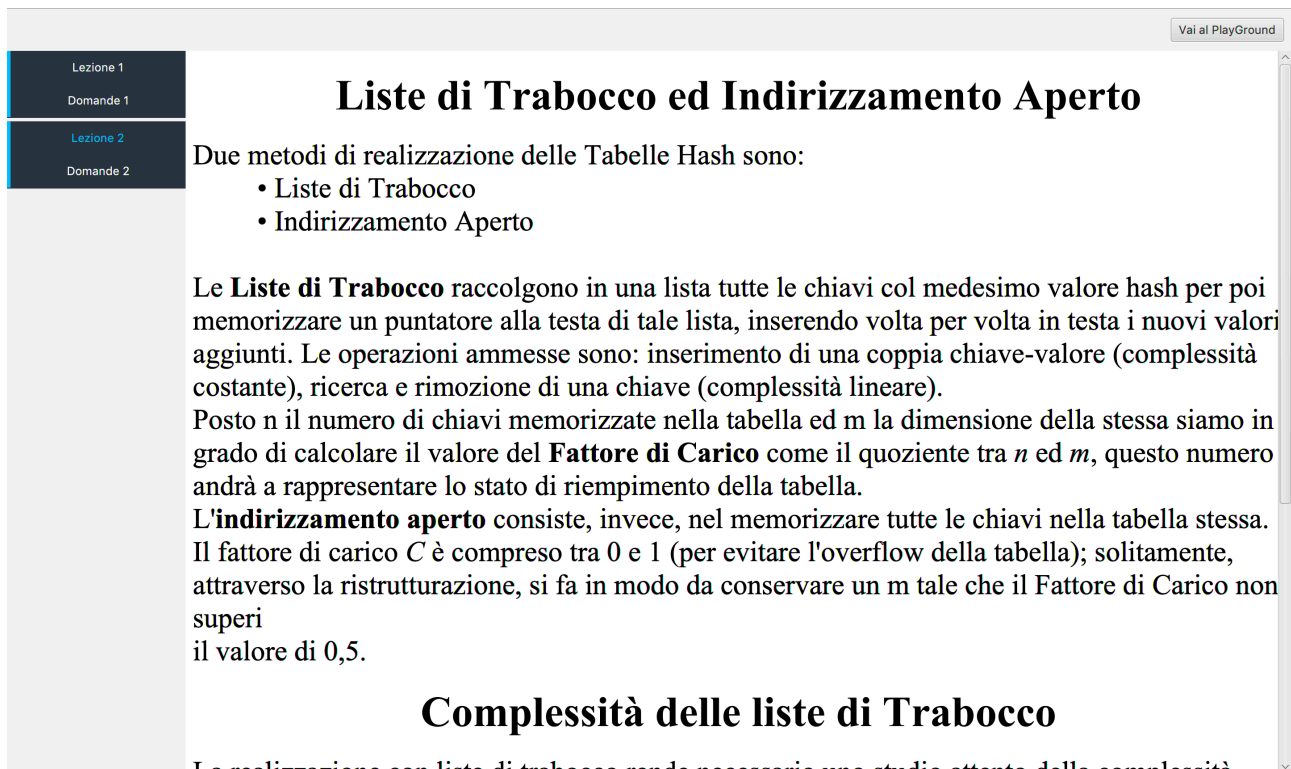


Fig. 1: Interfaccia della sezione lezioni

Playground

L'interfaccia del Playground è formata da 4 componenti principali:

1. La **Toolbar** (in verde in Fig. 2), che contiene i pulsanti per gestire le animazioni della tabella (compreso quello che consente di abilitarle o disabilitarle), quelli per effettuare inserimenti e rimozioni e quello per tornare alla sezione delle lezioni.
2. La **Configuration Bar** (in rosso in Fig. 2), che permette la configurazione completa della tabella.
3. La **Log Bar** (in azzurro in Fig. 2), che contiene un TextArea all'interno del quale vengono mostrati i messaggi di errore relativi alla tabella, come ad esempio quello che informa l'utente quando la tabella è piena.
4. Il **Viewer** (in arancione in Fig. 2), che si trova al centro dell'interfaccia ed è dove la tabella viene mostrata.

Il più importante di essi è ovviamente la Configuration Bar poiché permette di modificare molto finemente il comportamento della tabella. I parametri principali di configurazione sono infatti i seguenti:

- **Capacità Massima**, ovvero la capienza dell'array che contiene le coppie chiave-valore della tabella. Per motivi sia implementativi che di buon senso non può essere maggiore di 20 poiché altrimenti JavaFX impiegherebbe troppo tempo per renderizzare la tabella e la visualizzazione della stessa all'utente risulterebbe scomoda (come si può notare dalla Fig. 2 infatti già con una capacità di 10 parte della tabella risulta nascosta).
- **Funzione Hash**, la cui scelta comprende il metodo di estrazione, quello di divisione, quello di moltiplicazione e l'Xor oltre che la funzione Naive che distribuisce le chiavi in ordine alfabetico sulla base dell'iniziale.

- **Metodo di Scansione**, che permette di determinare la sequenza di ispezione della tabella in seguito a collisioni e la cui scelta si articola fra il metodo lineare, quello quadratico e l'hashing doppio.

A questi poi, a seconda del metodo di scansione scelto, ne posso essere aggiunti altri come il passo di scansione per il metodo lineare e per quello quadratico e la seconda funzione hash per l'hashing doppio.

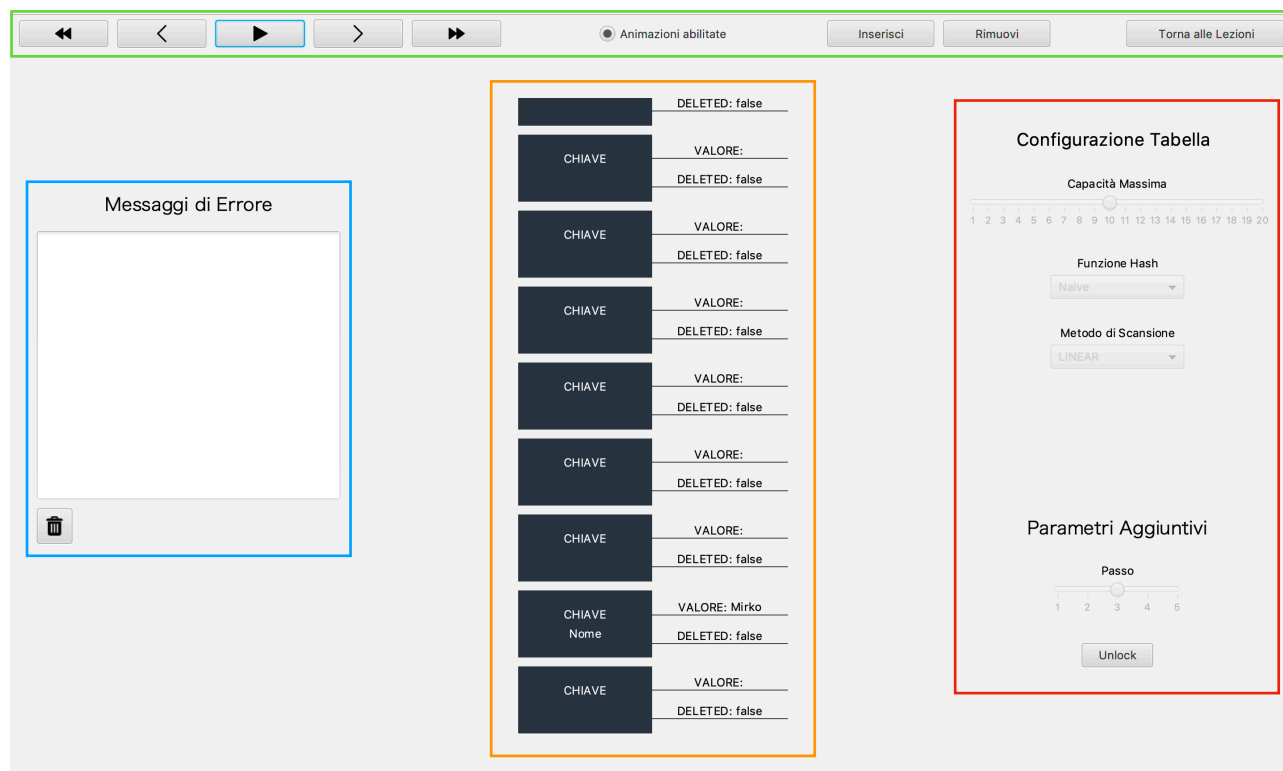


Fig. 2: Interfaccia del Playground

All'apertura il Playground mostra un dialog che permette di specificare la configurazione iniziale della tabella (che viene quindi riportata all'interno della Configuration Bar) ed eventualmente una serie di dati iniziali coi quali riempire parzialmente la tabella. L'utente fornisce un path ad un file .txt all'interno del quale i dati devono essere formattati in modo opportuno (ciascuna riga del file è vista come una coppia da inserire nella tabella e deve essere formattata nel modo seguente "chiave|valore") e AlgaT provvede automaticamente a caricarli ed effettuare gli inserimenti.

L'utente interagisce con la tabella principalmente modificandone la configurazione e utilizzando i pulsanti della toolbar per inserire e rimuovere chiavi ed eseguire le animazioni. Inserimenti e rimozioni si articolano in 3 passi fondamentali.

In primis l'utente configura la tabella attraverso gli appositi elementi dell'interfaccia.

Successivamente clicca il pulsante corrispondente all'azione che vuole effettuare e specifica, all'interno del dialog corrispondente, i parametri necessari. Oltre all'apertura del pop-up il click del bottone ha un'altra importante conseguenza ossia il blocco della Configuration Bar e la successiva rigenerazione della tabella con la nuova configurazione.

Senza questi passaggi la tabella finirebbe ben presto in uno stato indefinito poiché ciascuna chiave potrebbe essere stata inserita con una configurazione differente e risultare quindi irreperibile.

Infine avviene l'esecuzione vera e propria dell'azione richiesta sia essa un inserimento o una cancellazione. Questa, a seconda della preferenza espressa dall'utente, può avvenire immediatamente, ossia attraverso un semplice aggiornamento in background dello stato della tabella, oppure in seguito all'esecuzione di un'animazione. Siccome il secondo caso costituisce un'estensione del primo ci limitiamo a descrivere quest'ultimo.

Per entrambe le azioni possibili il primo passo consiste nella generazione della sequenza di ispezione sulla base del metodo scelto. Questa, che altro non è che un array di interi corrispondenti agli indici della tabella che verranno visitati, viene poi usata per “percorrere” la tabella e, se possibile, individuare la cella in cui effettuare la modifica. Si individua quindi una sotto-sequenza della sequenza d’ispezione completa il cui ultimo elemento costituisce per l’appunto la cella da modificare. La scelta di lavorare su una sequenza predeterminata invece che cella per cella è legata alla gestione delle animazioni. Questo consiste essenzialmente in successivi cambiamenti del colore di sfondo degli elementi ispezionati e pertanto conoscere a priori la sequenza delle celle da visitare semplifica notevolmente il lavoro.

Le animazioni inoltre sono rappresentate da una classe Java che estende la classe Transition di JavaFX e riesce così a sfruttare la gestione nativa delle animazioni presente all’interno del framework. Ad animazione completata se la modifica è possibile viene effettuata altrimenti il messaggio d’errore opportuno (chiave inesistente, tabella piena, ecc...) viene comunicato all’utente all’interno della Log Bar.