RELAZIONE PROGETTO “AlgaT”

INTRODUZIONE

Lo sviluppo di questa applicazione è stata la nostra prima esperienza di programmazione di un progetto completo e di queste dimensioni; partendo con una conoscenza di Java derivata unicamente dal corso universitario la scelta su come strutturare il progetto è stata attentamente ragionata, questa è stata sviluppata in modo da essere flessibile, non vincolata all’argomento scelto e alla quantità di contenuti.

La scelta finale è stata quella di utilizzare JavaFX unito a FXML e conseguentemente Scene Builder, le motivazioni iniziali erano principalmente dovute all’apparente semplicità dell’ambiente di sviluppo; l’IDE utilizzato è IntelliJ IDEA unito al JDK 11 e la release Gluon di JavaFX 11.

STRUTTURA PROGETTO

Il progetto è composto da 6 classi in totale: la classe contenente il main, 4 classi controller e una classe utilizzata a database. La classe database è fondamentale in quanto oltre a contenere le informazioni testuali è utilizzata per memorizzare la sessione corrente e scambiare dati tra le classi controller.
Le classi controller gestiscono rispettivamente 4 strutture FXML: la home, il pannello adibito alla parte testuale, la sezione di autovalutazione e la visualizzazione interattiva della struttura dati.

Tutte le scene FXML sono realizzate a partire da uno StackPane, utilizzando la sua struttura ad albero i contenuti di ogni scena vengono “attaccati” alla primaria, in questo modo la scena principale non cambia mantenendo le eventuali dimensioni scalate e dando la possibilità di aggiungere animazioni alle transizioni.

Ogni classe controller oltre alle funzioni per cambiare scena (tornare alla home o altra sezione utile) ha un metodo “SetData” il quale si occupa di aggiornare il database utilizzato da tutte le classi, questo metodo contiene inoltre operazioni atte ad impostare correttamente la pagina alla sua apertura.

LE CLASSI IN DETTAGLIO

CLASSE DATABASE

Per poter realizzare una struttura flessibile ed espandibile è stato necessario memorizzare i dati esternamente, inoltre per rendere i contenuti non dipendenti dalla scena FXML e non gestiti da infinite condizioni le informazioni dovevano essere accessibili sequenzialmente; Database risolve queste necessità utilizzando degli array di liste: gli array sono inizializzati nel costruttore, per inserire lezioni in AlgaT è quindi necessario incollare testi titoli e domande nei rispettivi array. Database contiene anche array di booleani per memorizzare i progressi dell’utente nella sessione corrente; la parte più “ingegnosa” di questa classe è la gestione randomica delle domande dei quiz: il sistema utilizzato carica le opzioni in un’unica lista, l’opzione corretta è sempre scritta per prima, il costruttore crea poi un array di liste contenente valori randomici compresi tra 0 e 2, questi valori indicano la posizione della risposta corretta dei quiz e rimangono invariati per tutta l’esecuzione.

CLASSE HOMECONTROLLER

La classe HomeController si occupa della gestione della pagina inziale contenente informazioni di contesto e i bottoni per accedere alle altre sezioni; inizialmente è unicamente disponibile l'introduzione all'argomento, le successive sezioni sono bloccate e saranno rese accessibili proseguendo il tutorial.

La gestione dei bottoni è affidata al sopracitato metodo "SetData" che invocato da un'altra classe aggiorna il database e in base all'avanzamento corrente sblocca le sezioni successive e modifica le immagini di copertina. I metodi ulteriori gestiscono il cambio di pagina e sono invocati in seguito alla pressione del relativo bottone se attivo, entrambi prevedono anche la riproduzione di una leggera animazione. Il metodo "goToLesson" carica i contenuti relativi alla lezione selezionata, mentre il metodo "goToAnimation" carica la pagina con la visualizzazione della struttura dati.

CLASSE TEXTCONTROLLER

La classe TextController mostra le informazioni testuali relative alle varie lezioni, il suo metodo "SetData" carica il database e la lezione richiesta, i contenuti mostrati vengono quindi impostati; una piccola interfaccia con quattro bottoni permette di navigare tra le pagine fittizie in quanto è solamente il contenuto ad aggiornarsi, inoltre sono presenti i collegamenti alla home e agli eventuali quiz.

Per fare questo sono impiegati due metodi "nextPage" e "prevPage" che grazie a un contatore impostano i contenuti testuali in base alla pagina richiesta, viene inoltre mostrato un indicatore di pagina; le pagine scorrono circolarmente. Sono presenti anche i metodi identici "goToHome" e "goToQuiz" che caricano la scena selezionata (in "goToHome" viene chiamata "SetData" con i parametri dettati dallo svolgimento del quiz).

CLASSE QUIZCONTROLLER

La classe QuizController è più complessa poiché oltre a gestire i quiz dà un feedback sull'andamento della sessione: la domanda viene mostrata in un pannello con tre possibili opzioni e i relativi selettori circolari, è presente un bottone per controllare la risposta che mostra un altro pannello con spiegazioni, in aggiunta un indicatore a 5 quadrati mostra la progressione nel quiz e la posizione attuale evidenziando con il colore le risposte corrette o errate, persiste l'interfaccia di navigazione con i pulsanti di navigazione e i collegamenti.

I selettori di risposta le label e gli indicatori sono inseriti in Array, il metodo "initialize" si occupa in questo caso appunto di inizializzare gli array; "SetData" anche in questa classe imposta il database e i dati in base alla sezione corretta oltre a mostrare l'avanzamento corrente nel caso si fosse abbandonata la sessione; è presente il metodo "SetCorrect" che, utilizzando l'array con le risposte corrette in database, imposta il testo della label "corretta" a quello della prima opzione ed inverte le altre label.

i metodi di navigazione tra le domande sono implementati in modo analogo a quelli di TextController, ma hanno controlli aggiuntivi per svolgere il quiz nell'ordine corretto, in aggiunta a ogni domanda viene resettata la selezione, nascosto il pannello con la risposta e impostato correttamente l'indicatore. Il metodo "check" controlla la risposta data confrontandola con i dati in database, lo aggiorna e riporta il conseguente

feedback. I collegamenti alle altre classi sono gli stessi con la modifica per sbloccare la nuova lezione solo una volta terminato il quiz.

CLASSE ANIMATIONCONTROLLER

AnimationController gestisce il pannello con l'algoritmo visualizzato, l'animazione è basata su "frames" che memorizzati in database contengono immagini e informazioni testuali, questi frames vengono fatti scorrere dall'utente mediante un pulsante; "SetData" importa il database e chiama il metodo "reset" che si occupa di impostare tutti i contenuti al loro stato iniziale, il reset può anche essere chiamato dall'utente tramite un pulsante nell'interfaccia di navigazione; per eseguire l'animazione a passi è utilizzato "setNext" il quale incrementando un contatore imposta testo e immagini al frame successivo.

INTERAZIONE UTENTE

AlgaT non contiene nessun warning o messaggio di errore, l'applicazione è stata programmata con l'intento di essere intuitiva, l'utente è libero di muoversi al suo interno, ma senza compiere determinati progressi l'avanzamento rimane sospeso. La schermata inziale mostra 4 bottoni di cui 3 bloccati, dopo aver letto l'argomento introduttivo è possibile passare alla lezione successiva, terminata la lettura è però necessario svolgere tutto il quiz per sbloccare la parte seguente; iniziato il quiz è possibile tornare alla home o rileggere il testo, ma non è concesso visualizzare la domanda successiva se non si è risposto correttamente alla corrente, nel caso l'utente abbandoni il quiz una volta riaperto i progressi rimarranno salvati. Terminando la prima sessione di quiz si sblocca anche la visualizzazione dell'algoritmo che aiuta l'utente nella comprensione dell'ultima lezione; ultimato anche il secondo quiz l'applicazione resta completamente navigabile: è possibile leggere tutte le sezioni e rivedere i quiz con le spiegazioni delle risposte corrette.