Controllare sta libreria nativa del JDK per gestione file audio: <https://docs.oracle.com/en/java/javase/17/docs/api/java.desktop/javax/sound/sampled/package-use.html>

LETTORE MP3 - 15/02/2023

Mail dei partecipanti:

* [alex.mazzoni3@studio.unibo.it](mailto:Alex.mazzoni2@studio.unibo.it)
* g[iovanni.luca@studio.unibo.it](mailto:Giovanni.luca@studio.unibo.it)
* s[amuele.casadei3@studio.unibo.it](mailto:Samuele.casadei3@studio.unibo.it)
* matteo.zoli7@studio.unibo.it

Progetto scelto:

Abbiamo deciso di creare un Lettore MP3, che gestisce le canzoni scaricate nel computer, ogni cartella è una playlist ed essa può contenere dei brani mp3. Pensiamo di utilizzare la libreria “JavaFX” per l’implementazione grafica e per la manipolazione di file audio.

Descrizione e motivo del progetto scelto

(Alex):

Mia madre dà lezioni di ballo a ragazzi disabili per un’associazione no profit. Essendo considerato in famiglia quello più bravo con i computer ero solito a passare i miei weekend scaricando musica e creando playlist, un compito che con il tempo divenne sempre più impegnativo, viste le numerose canzoni su cui lavorare.

In seguito ad una ricerca superficiale delle applicazioni di questo genere, mi sono reso conto che non esiste nessun lettore mp3 in grado di gestire i file e leggerli direttamente dal File System, perciò ho deciso di cogliere la palla al balzo proponendolo come progetto.

Pensavamo di focalizzarci nella resa di un’applicazione, come prodotto finito, avente un utilizzo semplice ed intuitivo per renderla il più user-friendly possibile.

Funzionalità obbligatorie:

* Implementazione grafica
* Importazione automatica dei file audio da una cartella specificata. (ogni cartella sarà una playlist e si potranno avere playlist che contengono playlist ma in questo caso esse non potranno contenere canzoni).
* Creazione e gestione dei brani (rinominarli e ordinarli).
* Creazione e manipolazione delle playlist.
* Controlli base per la riproduzione: Play/Pause, Prev/Next, Volume, Timeline.
* Riproduzione “Loop”, riproduce il brano corrente più volte.
* Funzionalità “Termina riproduzione”, dopo aver terminato la canzone non passa automaticamente a quella successiva ma termina.

Funzionalità opzionali:

* Gestione della coda di riproduzione (aggiungere brani in coda o eliminarli).
* Funzionalità “Cerca” per cercare un brano o una playlist.
* Funzionalità “View” per cambiare il modo di visualizzazione dei brani (per mostrare più informazioni di essi).
* Riproduzione casuale.
* Velocità di riproduzione.
* Suggerimenti dell’applicazione (Shortcut da tastiera).
* Semplificazione dell’utilizzo (ad esempio rendere semplice la navigazione della timeline delle canzoni, suggerimenti in base all’utilizzo dell’utente).
* Statistiche dell’utente usando I metadati dei file mp3 (canzone più ascoltata, genere, tempo totale di canzoni ascoltate, ecc.).
* Tagliare i brani (comprende anche ascoltare ogni punto specifico di un brano per consentire all’utente di tagliarlo in modo più preciso possibile).

Suddivisione del lavoro:

Alex: creazione base mvc, importazione delle canzoni e playlist, gestione della coda(con implementazione grafica), funzionalità “cerca”.

Christian: Controlli per la riproduzione base (con implementazione grafica), funzionalitá aggiuntive di riproduzione (velocità riproduzione, termina riproduzione), Statistiche dell’utente usando I metadati dei file mp3 ((tipo il wrapped di spotify) canzone più ascoltata, genere, tempo totale di canzoni ascoltate, ecc.).

Samuele: Creazione di canzoni(con implementazione grafica), Creazione delle Playlist(con implementazione grafica), semplificazione dell’utilizzo.

Matteo: Tagliare i brani(con implementazione grafica), Funzionalità "Loop”, funzionalità “view”, riproduzione casuale, shortcut da tastiera.

//immagini: Immagine che contiene testo, schermata, numero, diagramma

Descrizione generata automaticamente

