Project Requirements

ℹ️ **EECS 2311 W22 - Group 9** `v0.2`

Members:
- Vivek Wadhwani (*vivek121@my.yorku.ca*)
- Kris Singh (*ksingh7@my.yorku.ca*)
- Kingsley Okon (*King808@my.yorku.ca*)
- Lan Zhang (*zhalala8@my.yorku.ca*)

Project: github.com/devivekw/TAB2XML

Documentation Link

1 Introduction

`TAB2XML` is Java/Gradle based tool that enables users to convert musical tablature from text to MusicXML (*an open source standard for exchanging digital sheet music*) and enables them to play their converted file as well as visualize their file. This document outlines how a step-by-step guide to install and use the system for common use cases.

1.1 Scope

The intended audience and use for this application/document is mainly guided towards testers for this system, as well our customers. This should cover most of the relevant use cases for the most up to date version of our system, further changes will be reflected in this document.

1.2 Overview

This app is built to be able to run offline and the user can interact with a graphical user interface that is simple and ease to use. The application will also allow the user to import a file of their choosing and export into a MusicXML file as well as the ability to play, pause, forward and rewind their converted audio file.

2 Requirements

This section will detail in which capacity the application will be able to work and its intended functionalities.

2.1 Functional Requirements

The functional requirements of the application are the features present and what the user can do within the application.

- The user must be able to input valid guitar or drum tablature into the application.
- The user should be able to choose which features they wish to use through the push of a button.
- The user must be able to convert the provided tablature into sheet music.
- The user must be able to play the music of the tablature they provided.
- The user must have access to playback features such as play, pause and seek.
- The user should be able to view the converted music sheet with detail.
- The user should be able to change the beats per minute of the playback.

- The user must be able to include tablature to support repeated playback features.
- The user must be able to input a measure number which directs the user to the respective section on the sheet music.
- The user must be able to visualize the play functionality with the output as they start playback, the visual output should also be highlighted with respect to the note currently being played.

2.2 Non-Functional Requirements

The non-functional requirements of the application are less technical than their functional counterparts, but are important nonetheless.

- The application should be user friendly and easy to understand.
- There should be a user manual to explain the application to new users.
- The application should be well tested to ensure it works in the correct user scenarios.

3 Uses

As there exist requirements for this application, user stories and cases are important for understanding the requirements and the functionality of the application.

3.1 User Stories

Different user stories help the development team envision what the intended uses are for the application.

1. **Primary Actor:** Music Teacher
   Goal: The teacher would like to display a tablature for a song they would like their students to play.
    a. The teacher starts the application and inputs the tablature onto the platform.
    b. They then select the `Preview Sheet Music` feature to convert the tablature into readable sheet music for the students.
    c. In addition, the music teacher to be able to play the music for the students so that they have a better understanding of the song.

   d. If the students are continuously practicing, the playback feature will be useful as it will allow them to rewind back if they need to practice a specific part.

   e. As well as display the notes of the current tablature in sync with the song being played.

   f. The students are also able to go to a specific measure instantly if needed to speed up their workflow.

2. **Primary Actor:** Musician
   Goal: A musician who created their own tablature would like to easily convert their work into sheet music.

   a. The musician starts the application and inputs the tablature onto the platform.

   b. The musician will be able to enter their tablature and convert it into the desired sheet music. (*it automatically deciphers the needed instrument based on the input*)

   c. They will be able to listen to their song before actually attempting to play it in person, and can use the beats per minute functionality to determine how bpm affects their instruments playback.

   d. The musician is also able to playback the music sheet synchronously while playing back the tablature to provide a better visualization tool.

   e. The musician can quickly go to a specific measure if needed quickly to speed up their workflow and not hinder their creative process.

3.2 Use Cases

Similar to the user stories, the use cases will help the development understand the technical aspect of what the users should be able to do.

1. The user wants to play the provided tablature.

   a. The user launches the application and is greeted by the launch screen.

   b. They either paste in the tablature or import a text file containing it.

   c. The user clicks on the `Play Music` button.

   d. The application launches the Play Music window.

    e. The user clicks on the play button.

    f. The application plays the music until the user decides to press the pause button.

    g. The user can customize the bpm using the bpm slider which determines the length of the music being played.

    h. The user finishes playing and closes the application.

2. The user wants to preview the sheet music of the provided tablature.

    a. The user launches the application and is greeted by the launch screen.

    b. They either paste in the tablature or import a text file containing it.

    c. The user clicks on the `Preview Sheet Music` button.

    d. The application launches the Sheet Music window.

    e. The user is able to preview the sheet music for however long they wish while being completely in sync with the music playback so as to provide a better visual experience.

    f. The user is also able to highlight certain sections of the music sheet which they like and play those separately.

    g. Once they are satisfied, the user will close the application.