Allegro Tab Converter

System Requirements

Specification

Version 1.0 – February 28, 2021

Prepared By:

Rafael Dolores(216142069)

Juhnyeong Park(214199327)

Mohammed Fulwala(217459744)

Yashraj Rathore(216645814)

Shawn Verma(215064447)

Table of Contents

1. Introduction

This section introduces the purpose of the Allegro Tab Converter Application, the motivation behind it, and the specific group of audience it targets.

1.1 Background

Tablatures are continuously used by many musicians to assist them in understanding the rhythm and duration of a note within a musical piece. Due to their popularity and the difficulty it takes to read them, the music community has developed an open format called "MusicXML" that can denote a music piece precisely. The only issue is that there is no standard way to convert the tab into a MusicXML file.

The Allegro Tab Converter is a software that allows the user/customer to input a .txt or a simple text containing a guitar, bass, or drum tablature for a song and produces a .musicxml file that can be used for various purposes.

1.2 Objective

The objective of this report is to provide a comprehensive description of the expected system requirements for the Allegro Tab Converter. This document explains all the currently planned features of the software along its interaction with specific users. Furthermore, the designated functional and non-functional system requirements are also covered in this report.

_____

1.3 Intended Audience

The development of this software is dedicated to everyone in the music community. It can be used by anyone who wants to learn music, teach music, create music, and many others who want to use the features provided by this application for recreational purposes.

1.4 Product Scope and Constraints

The Allegro Tab Converter will be able to convert a guitar and bass text tablature into a MusicXML format. The application should be able to accept a tablature from the user if they use the interface's "browse" button and provide it from their file directory or they can copy and paste the tablature directly into the interface's textbox. As of this version, the app's feature only supports guitar and bass tablatures that contain regular notes. Any of the techniques below will not be recognized:

- h = hammer-on.

- p = pull-off.

- b = bend.

- / = slide up.

- \ = slide down.

- PM – – – – = palm muting (above or below TAB)

- ~~~ = vibrato.

- x = muted hit.

2 Primary Uses and Impact

This section describes the use of this software to our target audience and several scenarios in which it can be used to suit their musical needs.

2.1 Use Cases

2.1.1 Creating Educational Resources

Primary Actor: Musical Instructor

Success Scenario: An instructor finds a music tab in text file format to show to her students but deems it to be beginner unfriendly. The instructor uses this software to convert the file into a MusicXML file and shares its advantages to the entire class while further encouraging them to use it. As a result, she was able to provide a smoother learning curve for her students.

2.1.2 Self-directed Learning

Primary Actor: Novice Musician

Success Scenario: A starting musician decides to enhance their musical knowledge in their spare time. They find the tablature of their favorite song - In the end by Linkin Park - and finds it hard to read that tablature. Desiring for an uncomplicated format, they use the Allegro Tab Converter to transform that text into a MusicXML file. Now, they can use the new

format to allow a software like MuseScore to play that music and understand its rhythm.

2.1.3 Content Sharing

Primary Actor: Music Composers

Success Story: A music composer wants to share a recently created piece, in western notation, to the music community. Understanding that people from other parts of the world cannot read western notation, he inserts the text file into the application to convert it into a MusicXML file and shares this converted format instead along with instructions on how to use this file correctly.

2.2 User Stories

❏ As a music instructor, I want my musical tabs to converted into a MusicXML file so that I can offer my students an environment that will maximize their learning opportunities.

❏ As a self-directed musician, I want the application to convert any complicated tablature into a MusicXML file so that I could learn my favorite musical pieces at a much faster rate.

❏ As a music composer, I want the application to be user friendly and easier to use so that I could efficiently share my music with the rest of the world.

3. Functional and Non-Functional System Requirements (FURPS)

3.1 Functionality

3.1.1 Capability

This application should accept a text file which consists of the music tablature of a guitar and bass tablature or the direct text of the tablature into a text box. It should be capable of converting this tablature into a MusicXML file and automatically storing it into the system in which the application is being used.

3.1.2 Reusability

The application can support various types of music tablatures and is run on a stable Java environment. The algorithm is developed with the intention of inherently recognizing the set various instruments and thus can be reused to cater additional requirements with ease. Furthermore, the application delivers a converted .musicxml file, which means there will always be a need for this application. This is because .musicxml files are widely used within the music industry.

3.1.3 Security

This application should not store any unnecessary data from the system. It should also not extract any data while the application is running as well as while it is not running, that has not been authorized by the user. Furthermore, this

application should not corrupt the text file, the MusicXML file, or any other file in

the user's system.

3.2 Usability

3.2.1 Human Factors

This application is user friendly. If the user has not browsed any file, then the

user should not be able to convert anything. The copy and paste the text file to

the textbox's location. Moreover, the user should also be able to put in a direct

text from the tablature and be able to convert it to a MusicXML file.

3.2.2 Aesthetics

The interface for the application is designed intuitively for all the users to have

access to the application and use the application with ease. The application is

focused on the functionality without any unnecessary functions for the users.

3.2.3 Consistency

The application supports .txt tablatures for multiple instruments. The application

does not allow the user to convert any other files other than files in .txt format and

it warns the user. The application warns the user when conversion fails due to

errors in the tablature in .txt format.

3.2.4 Documentation

A user manual of the software will be provided to the target audience as soon as the project reaches its completion phase. The code will consist of clear and concise documentation using the JavaDocs feature.
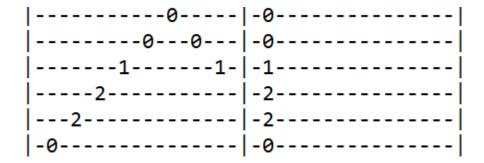
3.3 Reliability

3.3.1 Availability

This application runs on a Java SE runtime environment and must have it installed on the device to run. The application is supported for all types of OS and it requires approximately 30 MB for installing the application and to run the application.

3.3.2 Failure Rate & Duration

The failure rate of the application solely depends on the .txt file that the user wants to convert. As long it is a .txt file and its content are syntactically similar to that of a regular tab[1], the failure rate of the application remains minuscule.

```
|-----------0-----|-0---------------|
|---------0---0---|-0---------------|
|-------1-------1-|-1---------------|
|-----2-----------|-2---------------|
|---2-------------|-2---------------|
|-0---------------|-0---------------|
```

[1]

3.3.3 Predictability

The application returns a tablature in .musicxml format upon successful run. It will return with a warning to the user when a file that is not in .txt format is inserted or when the conversion fails due to errors within the .txt file.

3.4 Performance

3.4.1 Speed

The average runtime of the conversion must be less than 5 seconds depending on the size of the .txt file that the user desires to convert.

3.4.2 Efficiency

Assuming that the tablature in .txt format is written is syntactically similar to that of musical tab that the software supports [1], the application will run successfully.

3.4.3 Resource Consumption

The idea of the application is that it accepts a .txt file or a raw text containing the music tablature and creates a MusicXML file. The loading of the .txt file or the raw text into memory is the only resource that should be consumed during the process.

3.4.4 Scalability

The use of the application is fixed for tablatures in .txt format and only returns. musicxml file. It may support more tablature for different instruments upon development.

3.5 Supportability

3.5.1 Testability

The application returns warnings every time an error occurs within the expectation. It does not return any detailed description on the reasoning of the failure.

3.5.2 Extensibility

This application may support more instruments and convert in the manner of the user's choice upon development.

3.5.3 Serviceability

This application is a desktop application that can be run on all desktops with either Mac OS or Windows as the OS that has Java SE runtime environment installed.

3.5.4 Configurability

This application does not support any configure options to the users as it may

give the user a non-intuitive experience.