Requirement documents

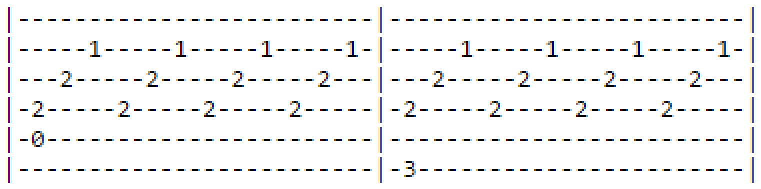
Team 1 – Tab2Pdf

Introduction

This document specifies the required features and input to output conversion of the tablature-to-pdf conversion software. The sections covered include: Brief Requirements, Acceptance Cases, Input Handling and Graphical User Interface (GUI)

Brief requirements

1. Convert ASCII text file to a PDF file.
2. Allow customization of title, subtitle, spacing, and scaling.
3. The user is able to interact with the system using a Graphical User Interface(GUI). The GUI will allow users to customize the look of the PDF file and choose the input file to read as well as where to save the generated output.
4. The user is prompted to select an input file, here a .txt file or a .tab file are acceptable inputs, as the guitar tabs will be written in ASCII format and text files are the best at keeping desired form of a guitar tab. If the program is asked to handle a file that is not formatted as a guitar tab then the program skips over it and prints a message to the user that the input file is invalid. A valid guitar tab text file uses '-' characters to represent fret lines and they are grouped in 6 rows, '|' characters represent bar lines and numbers represent the notes that will be played. The pipes should be arranged in the same column row by row and the notes will be arranged column by column to indicate when to play them and on a specific row to indicate which string is played. An example of a valid tab file can be seen here:



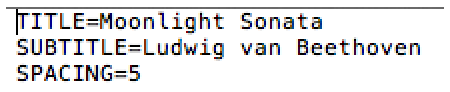
```
|-------------------------|-------------------------|
|-----1-----1-----1-----1-|-----1-----1-----1-----1-|
|---2-----2-----2-----2---|---2-----2-----2-----2---|
|-2-----2-----2-----2-----|-2-----2-----2-----2-----|
|-0-----------------------|-------------------------|
|-------------------------|-3-----------------------|
```

Conversion requirement

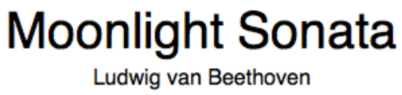
The section below is a list of cases where if certain characters or sequence of characters are found then they will be considered valid by the program and output as shown below. However, if it is not found valid then the program skips over that character or a sequence of characters. In each example below the input and output of the cases are shown.

Title, Subtitle and Spacing

The TAB2PDF program will look for the keywords "Title=" "Subtitle=" and "Spacing=" and take the value next to them so that they can be outputted into the pdf. The program can read these keywords regardless of the case. An example of how the title and subtitle are represented on the PDF can be seen below. If the title or the subtitle is not given then the program assumes that there is no title or subtitle leading to no title or subtitle in the output file. However, this can be edited in the GUI, description for that is provided below. Also, the spacing which determines how many bars can be in the width of a sheet can also be edited or provided in the GUI.



```
TITLE=Moonlight Sonata
SUBTITLE=Ludwig van Beethoven
SPACING=5
```



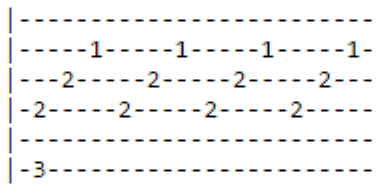
Moonlight Sonata
Ludwig van Beethoven

Arrangement of tabs:

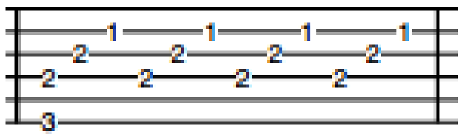
The program will be able to read line by line of the ASCII input text provided by the user and provide an accurate representation as a PDF file. Therefore, the program will always be able to read every line in the text file provided by the user, so no lines are ignored. Also, if the space is insufficient to fit a full tab on a line the program will move the individual bars to the next line to make it easier for the user to understand the music.

String lines:

Use dashes to help you understand the location of fret notes. By using dashes you can create columns to understand when to play each note and the order of playing. By aligning the fret numbers along the dash entries, the program can determine when the user intends to play the notes. The location of notes in the input file will be saved and converted to be in the same location on the frets. The program will change these dash symbols into solid string lines on the PDF making the tabs more readable.

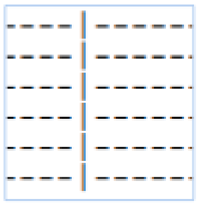


```
|-------------------------
|-----1-----1-----1-----1-
|---2-----2-----2-----2---
|-2-----2-----2-----2-----
|-------------------------
|-3-----------------------
```

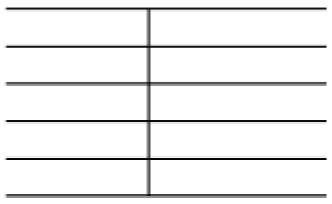


Vertical Bar Lines:

The program will read the text file and find where the vertical bar "|" lines are located, this allows the user to easily represent where a new bar starts. From which, there will be a solid bar that results in the output.

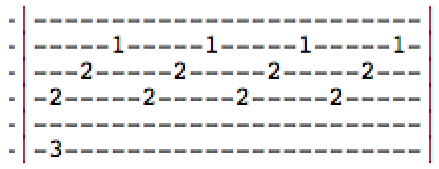


```
----|------
----|------
----|------
----|------
----|------
----|------
```

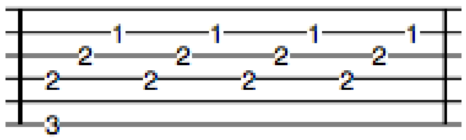


Fret numbers:

The program will read all single or double-digit numbers from the input file within range of 0 - 24, these digits represent the fret numbers of a guitar tab. All numbers are outputted in the same order as they appeared in the input. By simply placing the frets aligned in the same column, the user can represent when two strings are played at the same time and by adding horizontal space in between notes it can be distinguished what the next note to be played is. This input formatting is maintained in the final product and provides a traditional music notation for the tabs. When it comes to frets that are two digits in length the fret number will take up the space of two dashes and the digit will be separated from the next note by one dash.

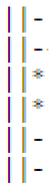
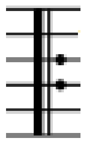
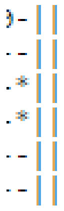
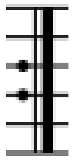
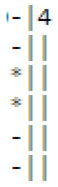
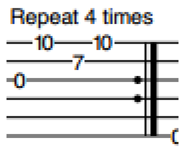


```
-|--------------------------|
-|-----1-----1-----1-----1-|
-|---2-----2-----2-----2---|
-|-2-----2-----2-----2-----|
-|--------------------------|
-|-3------------------------|
```



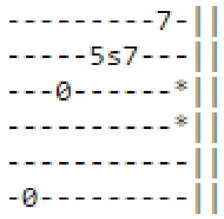
Repeats:

The program will allow the user to input tab repeat bars in multiple formats and convert them to a standard musical notation in the PDF output. This will notify the user on what kind of repeat is specified in the music and how many times a repeat is necessary. By formatting the input file as shown below, the program makes it easy for the user to represent repeats as in traditional music notation. The types of repeats include:

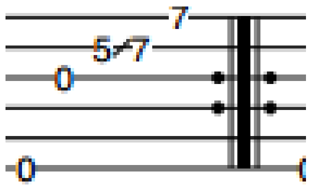
Tab Repeat Type	Sheet Music Repeat
	
	
	

Slide to and from fret numbers:

The user can represent a slide by inputting a character “s” in form “XsX”, where X represents a valid fret number.The output will contain a character similar to “/” in the same location where the “s” was located. The program will also allow the user to have any amount of “-” in the “XsX” and still produce the same output.

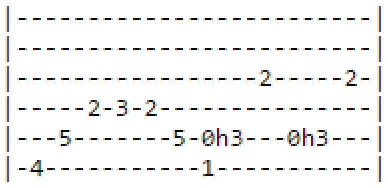


```
----------7-||
------5s7---||
---0-------*||
-----------*||
------------||
-0----------||
```

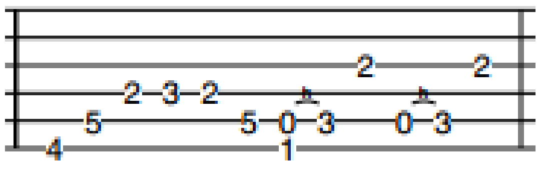


Hammer-on, Pull-off:

The program supports the creation of hammer-ons and pull-offs. When there is a “xhx” in the input, where x is a digit that represents a fret number, the program will write the fret numbers normally with an arc over them. The same is true for pull offs but has an input of “xpx” and will print the p character above the arc.The program will also allow the user to have any amount of “-” in the “xhx” or “xpx” and still produce the same output.

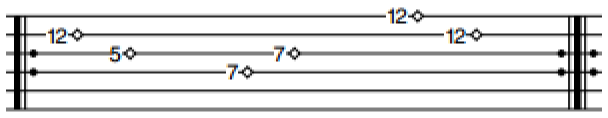


```
|-------------------------|
|-------------------------|
|-----------------2-----2-|
|-----2-3-2---------------|
|---5-------5-0h3---0h3---|
|-4-----------1-----------|
```



Harmonics:

The user can represent harmonics to fit the standard music notation by placing symbols denoted as <x> within the input file, where x represents a valid fret. The output is then the given fret number with a small diamond character to the right of the fret digit.

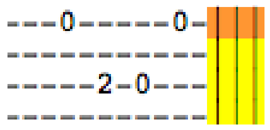


Muted String:

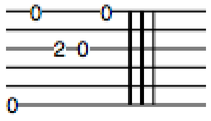
The input of the text file can contain a muted strings by an ‘x’ in the input where it is desired for a muted string to be played. The output file will also have a ‘x’ in the same location.

Triple Bar:

The program will let users represent triple bars in their tabs by placing “|||” lines in their input. After which the triple bar lines will be inserted into the PDF according to the location and spacing in the input.



```
---0------0-
------------
-----2-0----
------------
```



Graphical User Interface (GUI)

The GUI will be designed to be easy to use and allow immediate visualization of the desired output. The GUI will allow the user to easily customize and change the input file they have chosen to handle.

Easy-to-use:

The GUI will be designed to be easy to use for people of all experience with computers. The user has to only choose a file for input and a location to save the output file. If the user chooses not to customize any other features within the GUI, then the program assumes those specifications are already in the text file or the program assigns a default value for the spacing, while title and subtitle will be left blank. The GUI will allow the user to either select the input file from the browse window or to type/ paste the path to the file. The user will also be able to browse the location they want to save the output file or to type in the path. The browse restricts the user to only open .txt or .tab input files and to save .pdf files as output. However, in the future we plan to implement a feature that will accept any file as an input, and display an error message if it is not an accepted file.

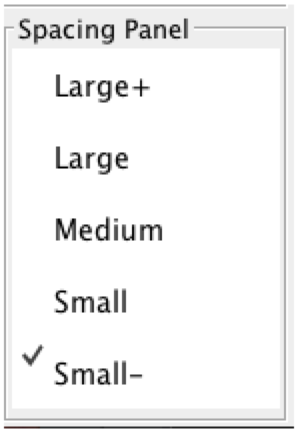
The GUI will also feature a preview of the text file on screen to allow the user to edit the file and then submit it to the program. The user will have a save button that will allow them to store all the changes they have made within the text box provided by the program. The user will be able to preview if the input file was read correctly.

The program will open up the created PDF file for the user to preview with the default PDF viewer. The user now has the opportunity to save the PDF if it is acceptable or continue editing the text file if the output is not satisfactory.

Customization:

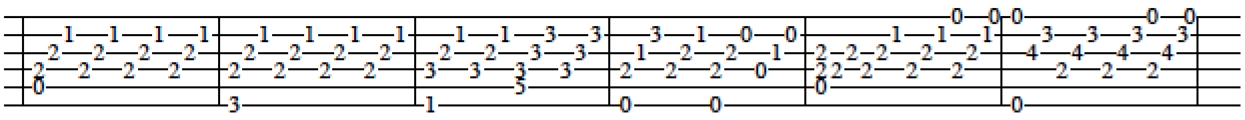
The customization options include changing the spacing and scaling factor using the buttons on the GUI. The user will also be able to change the title, subtitle and spacing value that is specified in the text file through the on screen text editor. The user will be able to control the appearance of the output file by changing the spacing within the GUI by a range of assigned sizes. The user will be able to alter any details within the document by using the on screen text box displaying the input file contents.

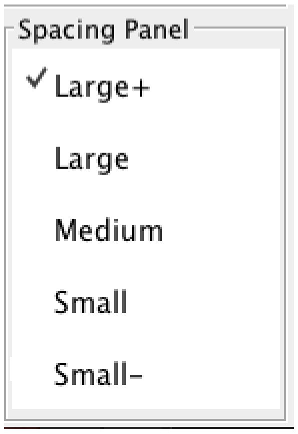
1. Spacing : Fitting a certain amount of tabs in one width of a page.



Spacing Panel

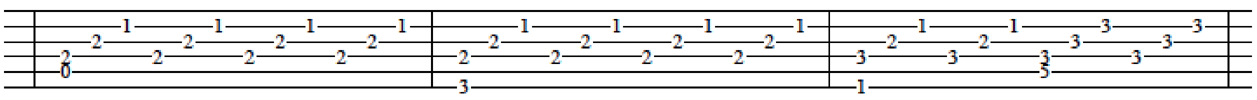
- Large+
- Large
- Medium
- Small
- [✓ Small-]



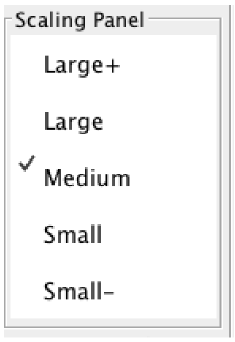


Spacing Panel

- [✓ Large+]
- Large
- Medium
- Small
- Small-

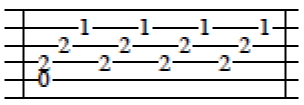


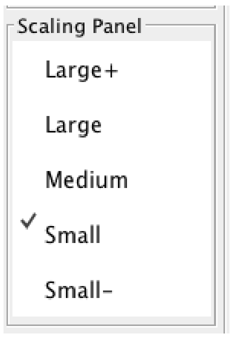
2. Scaling : The space inbetween the vertical lines.



Scaling Panel

- Large+
- Large
- [✓ Medium]
- Small
- Small-





Scaling Panel

- Large+
- Large
- Medium
- [✓ Small]
- Small-