CS5001 Final Project Report

Shang Xiao

May 2, 2022

1 Project Description

Chord Diagrams for Beginners is a mini-program that can teach beginners how to play basic chords of a specific song on their guitar. The program relies on three inputs from the user: original key, original key level, and target key. With these three inputs, the program will print out a chord diagram using Matplotlib to the user for each chord that wanted to know, as well as a new key that they want the current chord to be converted to, if they wish. The ultimate goal is to get everybody to enjoy guitar regardless of their musical background.

There are similar programs that are considered 'industry standard'. For example: GuitarTuna developed by Yousician is a tuning app that primarily focuses on guitar tuning for special string arrangements such as 'Drop D' or 'Open C'. These programs solve real-life examples and fulfill the needs of almost all levels of musicians, from beginner to advanced. The 'Python fretboard painter' project developed by Diego Penilla [1] has also inspired me to develop my own guitar program.

This project also has huge potential. For example, in later developments, I might consider integrating the most common chord progressions into the algorithm so that it can recognize more songs and convert them between different keys to fit the play style of more players.

2 Changes from Planning #2

2.1 Chord.py

Initial planning: Class Chord. All beginner chords will be stored in it. And we can easily extract a chord with particular key and level. The final version Class Chord stores the location of the fingers for each kind of chord. The load function calculates the corresponding chord by passing in the key and level, and then read the location of the fingers in the corresponding chord from the file. In additional to this, the original chord diagram was interpreted vertically, However, the arrangements for this diagram will affect the numbering within the '.txt' file which is used to store each finger position for each chord. Therefore, in the final version, this diagram is interpreted horizontally.

Planning #2:

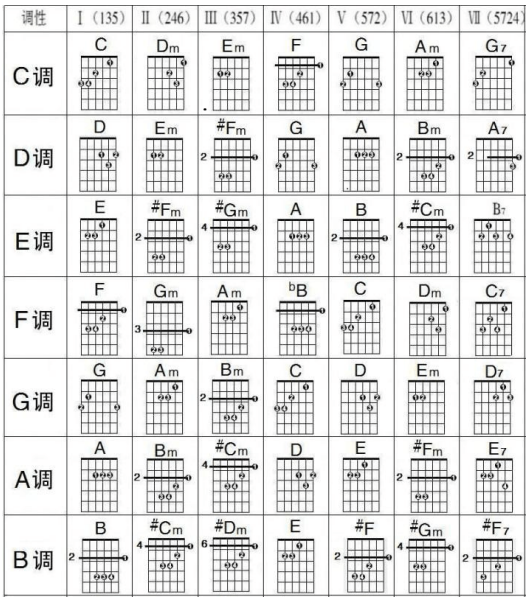

Figure 1: Planning #2

Final:

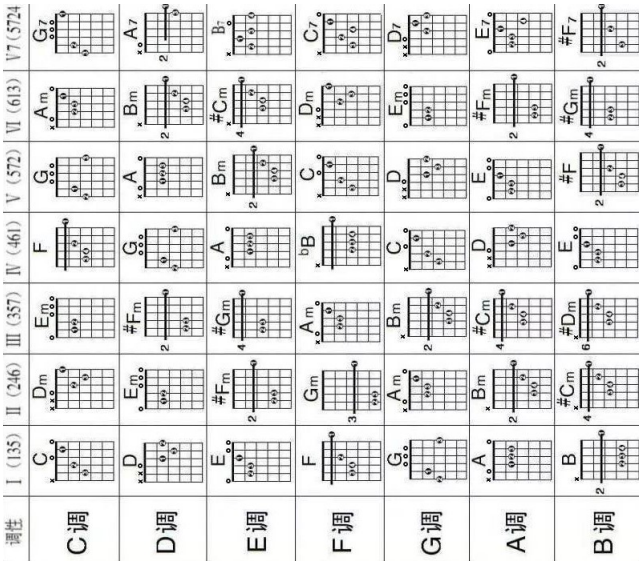

Figure 2: Final

2.2 FretboardPainter.py

Initial planning: Class FretboardPainter. This class is for guitar fretboard visualization using Matplotlib, so some basic parameters of visualization of Matplotlib will be stored in it. The final version Class FretboardPainter is to draw, the init function clears the board and draws the strings of the guitar. showChord function will draw the finger positions recorded in the chord in a graphical way, and show the Matplotlib interface. Chord progressions that contains arrangements between 2 to more key levels are not considered in this version for conversion to another key level due to complexity of combinations.

2.3 Draw.py

Since running a python file where there is only one pyplot provided by matplotlib, the Draw.py file is written. Each time main.py receives a command from the user it will record the command in the file and then call Draw.py to read the user's drawing requirements before making a single drawing.

2.4 main.py

The main function keeps iterating through the althrithm above as long as the user enters a valid key and level, the error is handled by the asking for permission method using if statement. The second part of the main function moves on to the converter which prompts the user to enter a new key level that they want to convert the current chord to.

2.5 test.py

Since this program heavily relies on user-input, most of the time, the correctness of codes and chord print-outs can only be verified after gathering user response (for example, to check if A 7 is E7 chord, we would first key in these values to enable Matplotlib print-outs). To test this program, I use a for loop to iterate through Chord.py to see if the output matches the data in each chord in folder 'chord'.

3 Reflection

During the final testing period, I can never get the E 5 - Bm chord displayed correctly, after checking the conditions in the load function I have found that the E 5 - Bm chord does not follow the pattern with the if statements and should be considered as a special case. It is clear how tiny imperfections can costs hours for me to figure out this issue and produce the final error version. It also shows the importance of testing in programming. Although in this case, the error does not involve any of the '.txt' file, I have found more than 2 incorrect finger position data within folder 'chord'. All of these corrections has definitely contributed to the final flawless version.

To reflect on the entire project, the final project definitely examines our overall programming skills mentioned in the planning 2 documents, which are conditional statements, loops, data structures, class, objects, and data visualization. [2] The completion process of this final project is also a final review of the entire course content for me.

Now I am confident to progress further into 5004 and 5008.

4 Resources

5.1 https://www.ultimate-guitar.com/ - Guitar tab for songs

https://www.fachords.com/guitar-alternate-tunings/ - Knowledge on Guitar tuning

https://betterprogramming.pub/how-to-learn-guitar-with-python-978a1896a47 - A much-advanced version of this project, my inspiration for this project comes directly from here. This project records each note when the user plays the note on their guitar once and recognizes it using the AAC recording Format to match the corresponding frequency (Hertz) of the note and convert them into the note on the guitar's fretboard. Super cool and interesting.

All other resources used has already been cited under each figure within this report.

References

[1] Diego Penilla. Learn guitar with python: Revealing the musical scales with lists, dicts, and matplotlib. May 2019. URL: https://betterprogramming.pub/how-to-learn-guitar-with-python-978a1896a47.

[2] PyPI. *PyGuitar: Chord book generator*, pyguitar 0.3.7 edition, June 2020. Maintainers: iwasakishuto, MIT License.