Python in the Enterprise

Course Project

PyChord

Basic harmonization for your own compositions

Team:
Izabela Bąk
Adrianna Pytel
Wiktor Jasiński
Szymon Telega

About application

Our application main purpose is to give the user complete music piece based on their own composed melody. It also allows them to check the chords of some existing songs, if only they can play it on the piano.

Nice keyboard, key choice, measure, volume, instrument choice, rhythm modification, sheet notes generation, saving the midi file and playing created piece - there is a whole set of functionalities the user may use to play with and have some fun, becoming a composer for a while.

The application includes the algorithm for harmonization, created and implemented by us, and the short description of it is given below.

Music Theory

Harmonization is the process of adding harmonies to a tune. In our application, the tunes are components of the melody entered by the user. Our basic harmonization takes each note separately and harmonize it within some musical rules.

Key

In order to have the harmonization possibility, user must choose the key of melody they wants to enter through the piano keyboard. This makes our harmonization operation slightly simpler to complete. Because our rules may be applied only for key-belonging notes, another ones are disabled for user (the buttons on keyboard become gray and not clickable). Otherwise, we would have to verify the key ourselves, which requires some more advanced methods, and may be tough for random melodies without no musical sense. Therefore if key is not chosen, the harmonization is not proceeded.

Rhythm

Our establishment to harmonize each note separately gives us some obstacles when it comes to the rhythm. User generally can change the rhythm of their melody by clicking on the note button, lengthen or shorten it. But when the note is too long to fit in the particular bar, it is moved to the next one. It eliminates the situation where notes are prolonged through the bar line, which might cause some problems with further harmonization (we couldn't apply some important rules to this form of melody).

Moreover, there are some further restrictions about the rhythm, associated with given above. Every ceaseless melody is valid, but the musical rests are yet not supported. Because of that, if entered melody have gaps, the user is invited to modify the melody in order to make it be continuous.

Algorithm

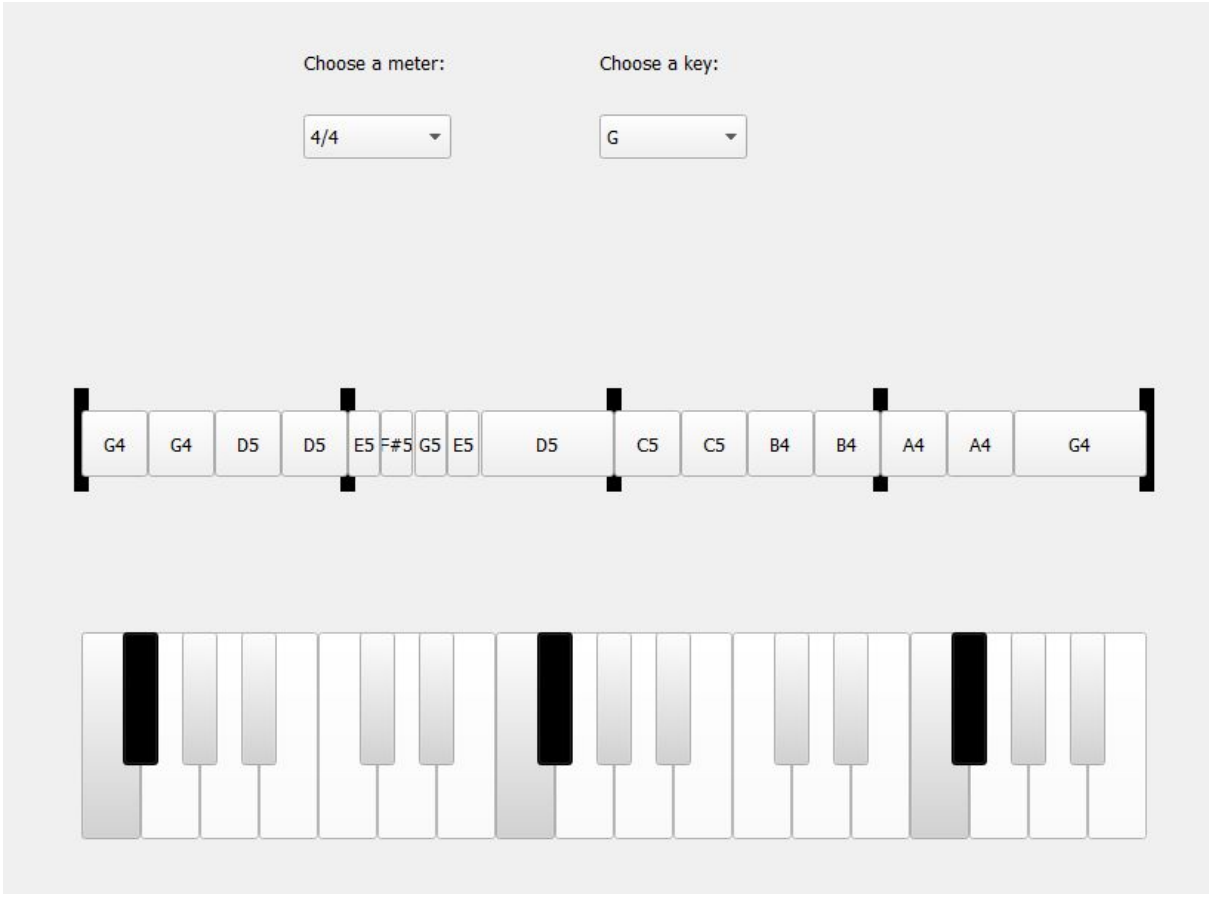
When the key is chosen and melody is valid, there comes the harmonization process. It is based on following rules:

1. For every note we have three options for chord choice - tonic, subdominant and dominant. They are respectively the first, fourth and fifth scale degrees. Each note may belong to one or two of mentioned (but when two, this is for sure the tonic-subdominant or tonic-dominant pair, this is because the subdominant and dominant have no notes in common).
2. For the first note we check the belonging of it in the following sequence: tonic, subdominant, dominant. It is nice to start the song with tonic chord, but when it isn't contained in tonic, we check subdominant or dominant.

3. Starting with the second note, we check which chord of those three it belongs to straightforward, with no order. If there is only one match, the matter is simple - the tune gets this particular chord and it is harmonized. But when we encounter two chords matching, we follow some additional rules:
    a. if the tune belongs to the same chord as the previous one too, this chord is maintained,
    b. if not:
        i. if previous chord was subdominant, it is time for dominant on the next note,
        ii. if previous chord was dominant, we need to resolve it musically by having the tonic for the next one,
        iii. if previous one was tonic, there is nothing to do here because we are sure that this situation is solved in 2a.

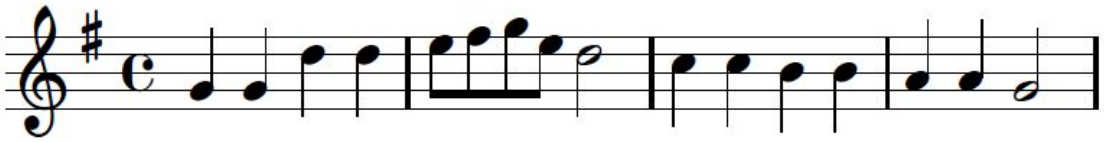
Algorithm example

Let's say our given melody is the well-known English song "Baa Baa Black Sheep". First step is to choose the key of G major. The meter is 4/4 by default. Then we enter first four bars as shown below:



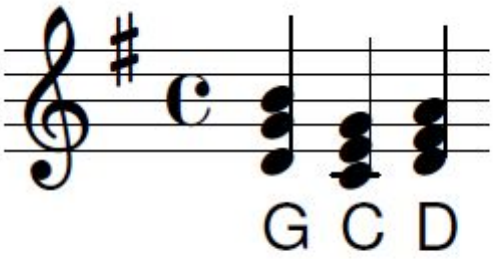
A we can see, in G key we can play only the key-matching notes. The ones which aren't allowed are disabled (with gray).

We can then show some notes:



In the key of G major, we need following chords that match our melody:

- tonic, which is G major (notes G B D)
- subdominant, that's C major (notes C E G)
- dominant, which D major (notes D F# A)

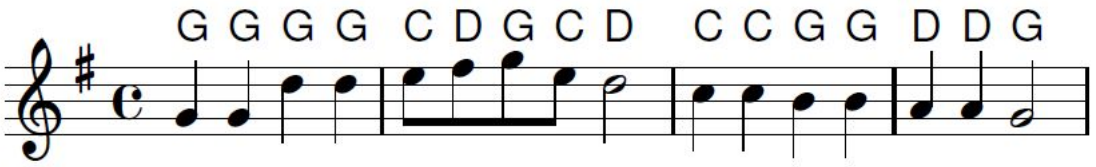


All of these three triads altogether contain all pitches in our key. There is no possibility to enter any pitch which belongs not to any of them. So, now we can successively match our notes to chords.

1. First note of the song is G. It fits to the tonic (G B D), which is nice for the first note, so we save the first chord as G major.
2. Then we have another G. Following the rule 3a, the G is maintained.
3. Next two notes are D D. They also belong to the tonic chord, so we have G major on whole first bar.
4. Then it's E. It belongs only to the subdominant chord. Our choice is then C major.
5. F# belongs to dominant. Simply we have D major on this note.
6. Then it's G. Now we have two options - tonic and subdominant. Following the rule 3bii, we put here the tonic chord, because it is our only possibility after the previous dominant. That's G.
7. Another C is for subdominant - C major.
8. The long D also belongs to two chords - tonic and dominant. But we know that after subdominant we rather want to hear the dominant playing, so the choice is simple (3bi), and we have the chord of D major here.

9. The second part of our piece is harmonized the same way. We have two C's in C major, two B's in G major, two A's in D major and we end the song with a really nice tonic, which comes after the dominant (case 3bii).

Our harmonized piece is shown below:



Further improvements

This algorithm is sufficient to perform the harmonization process compliant with the music theory. It may be extended by some more advanced stuff, including:

- more than three chords (submediant and maybe others),
- rests support
- minor keys
- harmonization independent of the rhythm (chords changing within one note or lasting for more than one, ligatures support)
- determining the key
- changing key within the melody