

HexaChord Extensions

Tim Clarkson, Yupeng Zhang, Michael Laporte

September 4, 2023

Abstract

This document outlines the specifications of the extensions to be made to the HexaChord software, a chord-mapping application used in computer-aided music analysis.

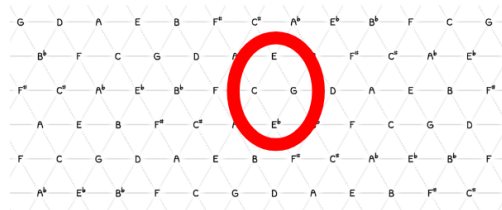
1 What is HexaChord ?

The software's general presentation is available on [Louis Bigo's page](#). The code is downloadable on the [project's Gitlab page](#). We have been working on the project on Linux and macOS. On Windows, we could compile but couldn't launch the project (we didn't want to spend time on that issue), so we do not recommend developing the app on that OS if this issue isn't your priority. If you have access to all of these OSes, macOS is the easiest to work with for this project !

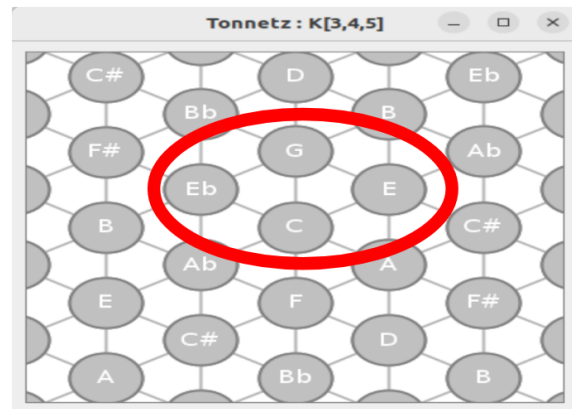
2 What has to be done ?

2.1 Step 1 : Change the grid

When the user opens HexaChord, loads as MIDI file and selects "display/hide complex", the displayed tonnetz doesn't follow industry standards. On the figure below, we have circled in red an area example from the tonnetz grid to show what is wrong :



Industry standard grid



What Hexachord displays

Figure 1: Comparison of the tonnetz grids

From the figure above, we see that to get to the industry standard grid from the grid displayed by HexaChord, two transformations have to be made :

1. Rotate the grid 90° anti-clockwise (aka trigonometric direction). This is already possible by right clicking on the grid and dragging the mouse, but it would be better to have the rotation done without the mouse action.

2. Mirror along the z-axis.

These two functions should be displayed in the control panel of the app to give the user more control over display.

2.2 Step 2 : Export the grid to video

Instead of using an external screen-recording-app, add an extension in the software that records what is being displayed on the grid. For example, have a "start recording" button and a "stop recording" button. Once the stop button is pressed, have the user choose name and file location of the video.

2.3 Step 3 : Simultaneously display the chords for at least two instruments

Right now, the software only displays the chords played by one instrument (piano) in yellow :

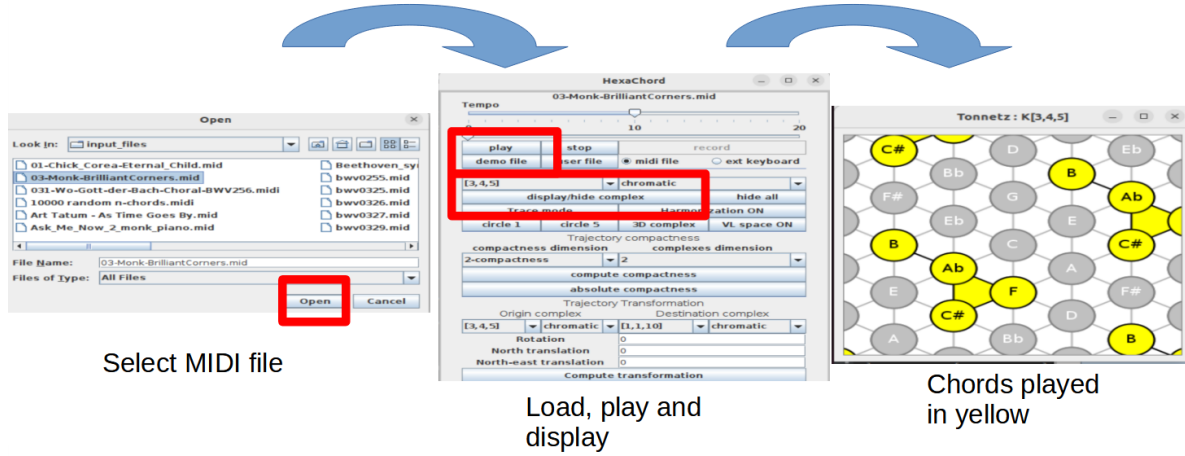


Figure 2: Highlighted chords from MIDI file in Hexachord

The goal of this step of the project would be to see when and where the chords played by different instruments visually overlap. To evaluate the relationship between two instruments (for example between piano in yellow and saxophone in blue), extract and display on the same grid the chords being played by the second instrument. At the moment, the user has to run the HexaChord animation, capture the screen with a different application, layout into adobe premiere and highlight manually the chords played by the second instrument.

2.4 Step 4 : Optimize the display of the chords

For this section, we need to know how much time it would take to optimize the display of the chords in this order :

1. Maximize overlap : don't just map out the second instrument like the first, but prioritise the display of pitches that instruments have in common as displaying them as "overlapped". This should show clearly any pitches that are shared by multiple instruments by occurring in the same location on the map, highlighted in each of the different instruments' colours.
2. Retain the existing algorithm in the software that minimises the distance on the map between sequential chords in a single instrument, but make this minimisation subservient to the overlap described in (1) for all subsequent instruments.

3 What has been done ?

Step 1 is almost finished. On the display, a 90° rotation is applied when the user clicks on the "rotate" button on the control panel or when the user presses "z" on the keyboard.

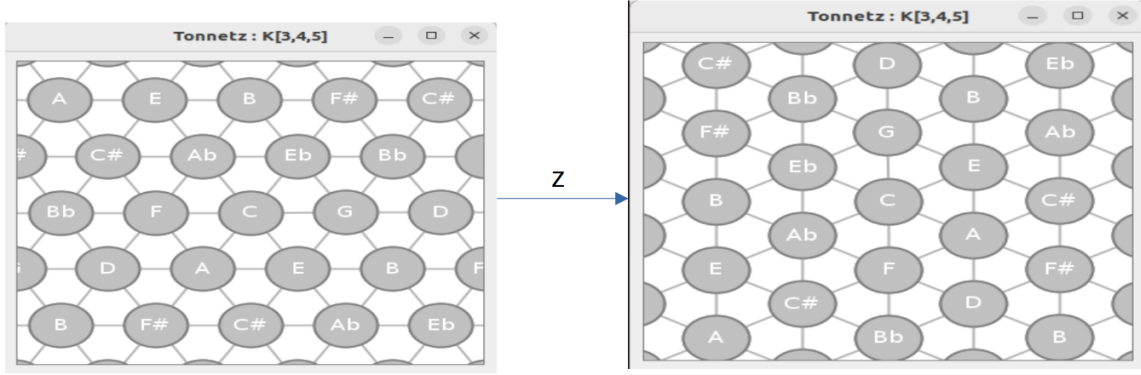


Figure 3: Tonnetz after clicking "rotate" on the control panel or pressing on "z" once (90° trigonometric rotation).

Mirroring is applied when the user clicks on the "mirror" button on the control panel or when the user presses "x" on the keyboard.

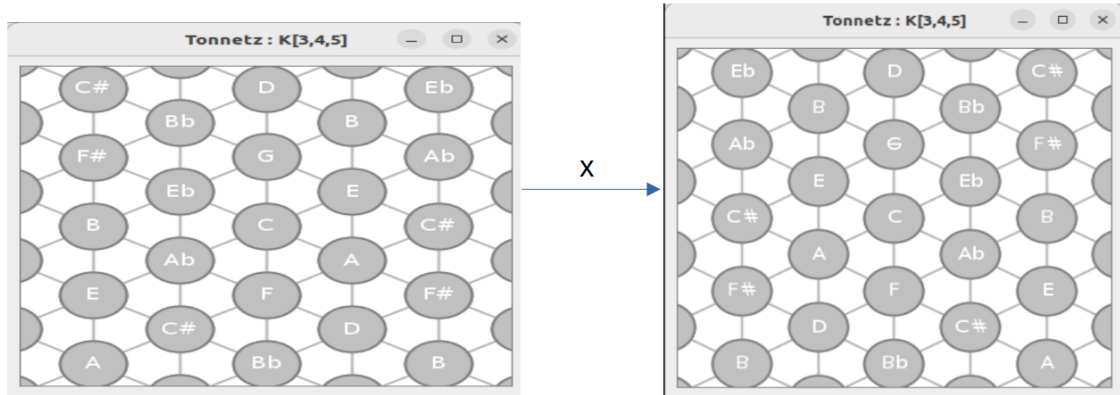


Figure 4: Tonnetz after clicking "mirror" on the control panel or pressing on "x" once.

There is however an issue with the displaying of the chords with the characters "#" and "G" as visible in figure above.

4 How has it been done ?

To help you quickly get your hands on this project, we will explain how we managed to achieve stage 1 goals in terms of modifications in the codebase.

Rotation and Mirroring: `src/Interface/JTransformPanel.java` & `src/Model/Music/Note.java`

Buttons addition: `src/Interface/InfoBox.java` & `src/Main/HexaChord.java`

5 What are the project's challenges ?

So far, we have spent some time on the following challenges :

- the project has almost no documentation (the academic papers linked to HexaChord don't detail the development of the software and there is no JavaDoc available).

- the code is poorly commented and there are many classes used in the project.
- M.Bigo (creator of the software) is rarely available.