Contents

1 zupfnoter

Work in progress: Web based editor for Tableharp notations based on abc notation

2 getting started

as of now the whole thing is far from being ready to use out of the box.

- install Ruby 1.9.3 or higher with bundler
- clone the repository
- goto 30_Sources/SRC_Zupfnoter
- run "bundle install"
- goto 30_sources/SRC_Zupfnoter/src
- run rake build
- open 30_sourcs/SRC_Zupfnoter/index.html in your webbrowser

3 Zupfnoter conventions in abc code

Zupfnoter tries to use ABC as close as possible. It does not add new syntax but applies to some conventions. These conventions reflect to

1. comments

   comments starting with `%%%%hn.` have a specific interpetation

2. annotations

   Annotations starting with one of `:`, `@`, `!`, `#` have a specific interpretation

The specific conventions in detail are as follows:

1. Jumps and repetitions

   This is done using anotations which is text in double quotes before a note. The target of the "jump" is denoted as "^:target", while the "jump" is dentoed as "^@target". Of course the same target can be part of multple jumps.

   You can control the position of the goto-line by adding a distance in halftones, e.g. "^@target@3", "^@target@-3"

2. Repetitions can also be controlled by chords on the right repeat bar. In this case target is left empty. For example

   ```
   "^@@-3" :|
   ```

   places the repetition line 3 halftones left of the end of the repetition.

3. Control visualization of Voices Synchlines, Jumplines, Flowlines

   This is done using specific comments with JSON syntax

   ```
   %%%%hn.print {"t":"all",         "v":[1,2,3,4], "s": [[1,2],[3,4]], "f":[1,3], "j":[1]}
   %%%%hn.print {"t":"all",         "v":[1], "s": [[1,2],[3,4]], "f":[1,3], "j":[1]}
   %%%%hn.print {"t":"sopran, alt", "v":[1,2],     "s":[[1,2]],       "f":[1],   "j":[1,2]}
   %%%%hn.print {"t":"tenor, bass", "v":[1, 2, 3, 4],   "s":[[1, 2], [3,4]],     "f":[1],  "j":[1, 3]}
   %%score (T1 T2)  (B1 B2)
   ```

   t The title of the print

v List of voices to be shown (it is an array of integer) from 1 to n denoting the voice index. Note that the voice index is basically the sequence of voices in the note preview. Therefore the %%score directive also influcnces the voice index.

s List of synclines to be shown. It is an array of array integers denoting the voice pairs for which synchlines shall be drawn.

f List of flowlines to be shown. It is an array of integers

sf
: List of subflowlins to be shown. It is an array of intenters. Subflowlines are flowlines connecting notes which otherwise have no corresponding note in other displayed voices and therefore would appear as single notes lost in space (without anny connection). startpos : the vertical position to start with the first note. It is an integer. j : List of jumplines to be shown. It is an array of integers l : List of voices to consider for vertical layout optimization. Defaults to the List specified by v

4. control the position of the legend

   ```
   %%%%hn.legend [10,10]
   ```

   where parameter is the legend position in mm from top left

5. augment the content of the legend

   The content of the legend is derived from the ABC metadata. You can append content to the particular lines by defineing an annotation with the same key. For example

   ```
   %%%%hn.annotation {"id": "K:", "pos": [-50,3], "text": "Original in F"}
   ```

   adds a note to the legend entry for "K:" which is the key of the music

6. sheet based annotations

   %%%%hn.note ["foobar", [10, 10], "large"]

   Parameters:

   1. Text
   2. position in mm from top left
   3. Textstyle "regular" | "large"

7. Note bound annotations

   1. you can define referrable annotations as

      ```
      %%%%hn.annotation {"id": "10", "pos": [-50,3], "text": "referenced annotation 10"}
      %%%%hn.annotation {"id": "11", "pos": [3,0], "text": "referenced annotation 11"}
      ```

   2. Note bound annotations are also entered as annotations, for example:
      `"^!Fine@10,10"` adds the word "Fine" at 10,10 mm from the note. Default position is 3,0
      `"^#10@10,10"` adds the content of hn.annotation with id: "10" (see 1.) at position 10,10 from note.

8. Lyrics

   Zupfnoter supports placement of lyrics by `w: lyrics` lines in ABC. You can control the position of lyrics by

   ```
   %%%%hn.lyrics {"pos": [50,50]}
   ```

4 Licencse

This software is licensed under dual license MIT and Commercial

5 Open issues

5.1 known bugs

001 Hightlighting in ace is turned off, since ace is throwing too many selection changed events 002 Play cannot be stopped 003 some refactoring necessary (see todo) 004 highlighting in tunepreview while playing does not work properly; tunepreview removes previous highlights 005 Q: tag is not considered while playing

5.2 current work items

101. drop down menu with proper links to informative sites
102. midi - play the generated harpnotes *done*
103. write messages to the console pane *done*
104. vertical resize of panes
105. zoom pan scroll in Notes pane *done*

5.3 User interface

201. zoom full screen of pane
202. cross-highlighting bewtween ABC (*done*) - Notes *done* - Harpnotes *done* - Player *done*
203. add a local description for ABC
204. add ABC-Syntax-Support to the Editor
205. minimize the panes
206. multilingual
207. incorporate bootstrap

5.4 More support for ABC

301. multiple staff / Voices to (support Bass harp) *done*

     better control about bass tenor alto soprano - requires certain refactoring *done*

302. annotations

303. trioles

304. ties and slurs

305. improved line handling: line break different between the voices ...

306. voice properties octave=...

5.5 Harpnotes

401. indicating measures *done*
402. vertical layout optimization (optimize the visual distance between two beats) *done*
403. annotations *done*
404. Debugging (writing the notenames in light grey) *cancelled*
405. draw extra flow line in unsynched notes (*cancelled*)
406. add marks to adjust the sheet in the harp *done*
407. print extracts *done*
408. configure vertical layout (fixed, optimized)
409. configure a transformation *done*
410. denote parts *done*

5.6 technology

501. MusicXml interface
502. Visualize the internal model for debugging purposes
503. Improved error handling

5.7 Player

701. Emulate harpplayer *in progress*
702. Metronome

6 Brainstorming

- using shoes and atom_shell to make a standalone application https://github.com/wasnotrice/shoes-atom

7 Result of initial evaluation

601. it is good to enter the stuff with two persons
602. good Visual feedback is essential
603. should be able to turn of some voices in oder to focus on the one currently entered
604. play from particular position onwards.

8