ZupfNoter Requirements

Bernhard Weichel (www.weichel21.de)

18.5.2014

Inhaltsverzeichnis

1 Introduction **3**

2 Objectives **3**

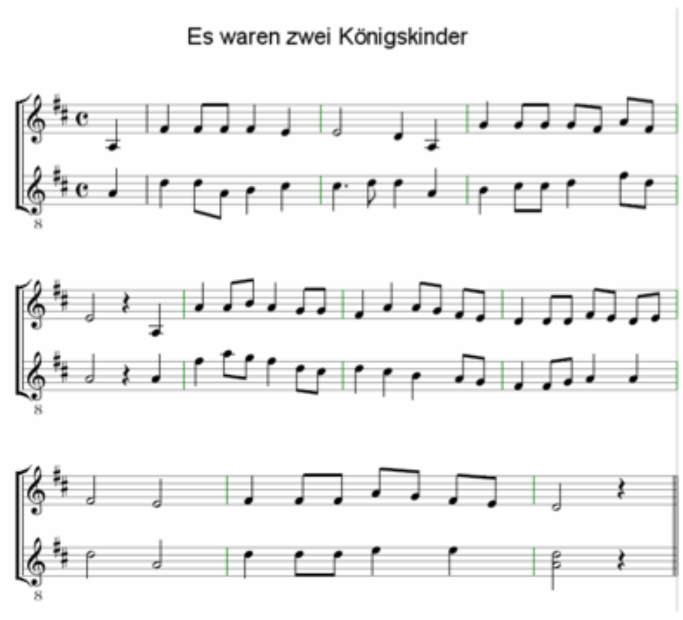
3 Requirements **4**

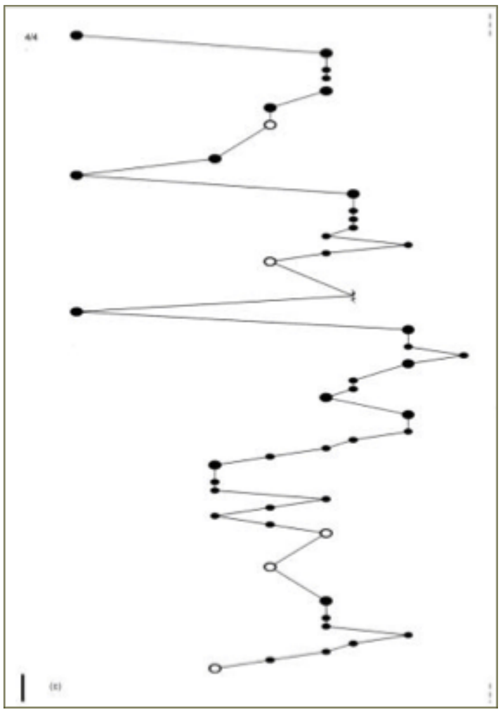
1 Introduction

Zupfnoter is an attempt to implemente a tool to create music sheets for “Table Harp” (http://Table Harp-harfen.de) respectively “Zauberharfe” (http://www.musik-im-spiel.de).

Basic approach in for these instruments is, that the music sheet is placed on the instrument such that the particular note and the string are visually connected.

The following example illustrates the approach.





This document describes the initial requirements for “Zupfpnoter”; Basic approach is

- use as commandline tool
- apply “abc-Notation and convert it to Table Harp Noten”

Details on abc see

- http://normanschmidt.net/abc/index.php
- abcnotation.com

Interesting details how to apply Table Harp see http://homepage.bnv-bamberg.de/flg-blw-partnerschaft/musik-facharbeit-eva-klein.pdf

2 Objectives

The following objectives are valid for Zupfnoter

- [RS_MG_001] **Easily create Table Harp Music sheets** { Zupfnoter shall be an easy tool to create music sheets for Table Harp. But it shall provide all the possibilites of the Table Harp - Notation. }()

- [RS_MG_002] **Apply for private use** { The primary focus of Zupfnoter is the private use. It is not intended for professional use. }()
- [RS_MG_003] **Open Source** { Zufpnoter shall be free and open source in order to help utilizatio of Table Harp. }()
- [RS_MG_004] **Utilize existing Software and Tunes** { There is a lot of music material which can be found e.g. in the internet. Such meterial shall be easily be used with Zupfnoter.

  Also there are many Typesetting Tools for music. Zufpnoter shall be utilized with such programs as well. }()
- [RS_MG_005] **Applicable on various platforms** { Shall be applicable on common platforms. }()

3 Requirements

3.1 System Requirements

- [RS_SYS_001] **Shall be portable** { Shall run on OSX, Linux and Windows. It should be implemented in a portable language.

  Example is java, python, ruby or javascript. }(RS_MG_002, RS_MG_003, RS_MG_004, RS_MG_005)
- [RS_SYS_002] **Shall be a framework easy to integrate** { Shall be a framework which can even be integrated in music editors

  As an eample see:

  - http://code.google.com/p/jspdf/

  }(RS_MG_002, RS_MG_003, RS_MG_004)
- [RS_SYS_003] **Shall apply popular frameworks** { Example for applicable Frameworks:

  - BytescoutPDF
  - jspdf
  - abcjs
  - drawthedots
  - prawnpdf
  - midjs
  - https://github.com/mudcube/MIDI.js

  }(RS_MG_004)
- [RS_SYS_004] **Interactive Operation** { Zupfnoter shall be operated as interactive tool. }(RS_MG_002)
- [RS_SYS_005] **Batch Operation** { Zupfnoter shall be operational as batch tool }(RS_MG_002)
- [RS_SYS_005] **Cloud storage** { Zupfnoter shall be able to communicate with cloud storage such as Dropbox, Owncloud }(RS_MG_002)

3.2 Input

- [RS_IN_001] **Shall be able to process various input formats** { Zupfnoter shall be able to process various input formats. Primarily such as

  - abc
  - MusicXml
  - Lilypond

  This might be implemented by predrive converters. }(RS_MG_004)

3.2.1 interactive editor

- [RS_IN_002] **interactive editor** { Zupfnoter shall be available as interactive editor such that one of the formats in [RS_IN_001] can be editied. Thereby an audible and visual feedback shall be provided.

  The kind of audible / visual feedback shall be selectable ([RS_IN_003], [RS_IN_005], [RS_IN_005])

  }(RS_MG_001, RS_MG_002)

- [RS_IN_003] **audible Feedback of current note** { The current note shall be played after it is entered / changed / selected.

  “Selected” means

  - that the cursor is moved around in the editor
  - A sequence of notes is selected in the editor
  - a note is clicked in one of the other visualizations

  }(RS_MG_001, RS_MG_002)

- [RS_IN_004] **Audible feedback to the entire tune** { The tune shall be played upon request:

  - entire tune
  - entire selection
  - entire tune starting at beginning of selection }(RS_MG_002)

- [RS_IN_005] **Visual Feedback as Music Sheet** { As the tune is entered / changed in the input field, a music sheet shall be displayed and updated. }(RS_MG_001, RS_MG_002)

- [RS_IN_006] **Visual Feedback as Table Harp Notes** { As the tune is entered / changed in the input field, a preview of th Table Harp Notes sheet shall be displayed and updated.

  }(RS_MG_001, RS_MG_002)

- [RS_IN_007] **Synchronized feedback** { The various representations shall be synchronized:

  - highlight the object in note sheet and “veen notes” when cursor is moved in input
  - highlight the the object in input upon click on a note in note sheet or “Table Harp notes” }(RS_MG_001, RS_MG_002)

- [RS_IN_008] **Automatic save and restore** { Zupfnoter shall be able to automatically save current changes in order to prevent from data loss. }(RS_MG_002)

3.3 Output of Harpnotes

- [RS_OUT_001] **Shall produce PDF output** { Zupfnoter shall produce an out in PDF format }(RS_MG_001, RS_MG_002)
- [RS_OUT_002] **Shall handle A4 pages** { A Table Harp notes are larger than A4 pages, Zupfnoter shall provide a print which can easily be combined.

  This is in particular to support the fact that private users usually do not have A3 printers [RS_MG_002]

  - divide the sheet on a4 pages
  - print proper marks where and how to glue the pages
  - print some overlap
  - print cut marks to indicate where to cut the second page
  - avoid cutting in between a note

  Note that smaller songs might fit on an A4 page in landscape. }(RS_MG_001, RS_MG_002)
- [RS_OUT_003] **Should support A3 printers** { There may be users with an A3 Printer, so it should support printing on a full A3 Page. }(RS_MG_002)
- [RS_OUT_004] **Optimize page layout** { Page layout shall be optimized depending on user request:

  compact the vertical distance is constant and selectable by the user in grid measueres. One grid measure is the size of the smallest note

  automatically the vertical distance depends on the amount of notes which need to be rendered such that the entire sheet is filled.
  In this mode, it shall be possible to select the voices to be considered during optimization. By this on can create excerpts where all voices share the same vertical positions.
  On the other hand, if only the printed voices are considered for optimization, it is likely that more music fits on the page.

  - }(RS_MG_002)
- [RS_OUT_005] **Shall allow to fine tune positions** { It shall be possible to control the position of

  - legend
  - annotations
  - harpnotes (beginning of the notes)

  }(RS_MG_001, RS_MG_002)
- [RS_OUT_006] **Shall allow to create excerpts** {

  for music with many voices it shall possible to print excepts e.g. soprano + alto

  }(RS_MG_001, RS_MG_002)
- [RS_OUT_007] **Shall allow page annotations legend and production notes** {

  It shall be possible to place

  - a page legend derived from the abc meta-data such as Key, Author etc
  - a production note indicating the version and host of Zupfnoter
  - arbitrary notes bound on page coordinates

  }(RS_MG_001, RS_MG_002)

3.4 Support of Table Harp notation

- [RS_HN_001] **Shall Support the Header annotation**{ Zupfnoter shall be able to denote the Meta-Information about the piece. It is a text block on the top left of the page.}(RS_MG_001)
- [RS_HN_002] **May Support the lyrics**{ Zupfnoter may be able to print the lyrics.

  Position of lyrics shall be under control of the user.

  }(RS_MG_001)
- [RS_HN_003] **Shall support Basic Table Harp notation**{ Zupfnoter shall support the basic Table Harp notation:

  - Repreesentation of note length

note	representation
full	big empty circle
half	empty circle
quarter	filled circle
eights	small filled circle
sixteenth	smaller filled circle or cross

  - The notes of one particular voice are connected by a solid line -[RS_VH_013]
  - Representation of rests

  }(RS_MG_001)
- [RS_HN_004]**Shall Support chords and polyphones**{ Zupfnoter shall support notes which are played simultaneously. This is indicated by dotted lines connecting the particular notes of the chord.

  Note that these polyphones might come from two different use cases:

  in Voice polyphones are part of one voice. Even if it is difficult to play there are use cases for this. These usually are played with two fingers of the same hand.

  cross voice in this case, multiple voices have a note start on the same beat. This might even happen in combination of “in Voice” polyphonse.

  }(RS_MG_001)
- [RS_HN_005] **Shall support repetitions**{ Zupfnoter shall be able to denote repetitions by drawing a rectangular line backwards.

  Also take care of nested repetitions

  }(RS_MG_001)
- [RS_HN_006] **Shall support variant endings** {

  Variant endings shall be supported such that

  - The beginnig of the variant ending is marked by an annotation

- The end of the variant ending is handled like a repetition }{RS_MG_001}

- [RS_HN_007] **Shall support arbitrary continuations** {

  It shall be possible to draw abrirary goto (also known as continuation, jump) lines e.g. for “ca capo al Fine”. For this purpose, it is necessary to markup the point where to leave the dentoed sequence as well as the point where to continue.

  This annotation shall be done using conventions in ABC code. }{RS_MG_001}

- [RS_HN_008] **Shall optimize the goto-lines** {

  goto-lines are rendered as

  - rectangular lines
  - start at after the note and end before the note. Therby the best side of the note shall be choosen.. E.g. if the target is northwest of the start, the line shall start at lower left corner of the start note and end at the top right corhner of the target.
  - vertical part of the line shall be controlled by the user

  }{RS_MG_001}

- [RS_HN_009] **Shall Support arbitrary annotations**{ Zupfnoter shall be able to denote short annotations embedded in the tunes. These notes shall be placed on the right part of the page vertically aligned to the respective note}(RS_MG_001)

- [RS_HN_010] **Shall support ties** {

  Ties are notes of the same pitch which are played legato. This is used if one wants to denote a sound which plays longer than the longest notifiable note. It also is used if e.g. a four beat sound is across measure boundaries.

  Support of ties means

  - respect tie when playing
  - mark the tie as a bow between the relevant notes.

  Ties are easy to draw since all involved notes have the same pitch.

  }{RS_MG_001}

- [RS_HN_011] **Shall support slurs** {

  In opposite to ties, slurs may connect notes of different pitch. Slurs are played “legato”. In fact on a table harp this not really a difference.

  When drawing a slur layout might be optimized according to the notes involved in the slur. Maybe an extra annotation can be capplied to manually optimize the layout of the slur. }{RS_MG_001}

- [RS_HN_012] **Shall support tuplets** {

  }{RS_MG_001}

- [RS_HN_013] **Shall draw melody lines** {

  Zupfnoter shall connect notes of the same voice as they represent a melody.

  It shall be configurable which voice represents the main melody.

  In particular, we see three kinds of melody lines

primary melody The main melody of the piece, usually “soprano”, Voice 1

secondary melody The second melody, e.g. “tenor”, voice 2

Melody lines shall be not drawn to the beginnig of a new part. see [RS_HN_016]

}(RS_MG_001)

- [RS_HN_015] **Shall draw orphaned melody lines** { Notes of a “non melody” voice which have no counterparts in an other voice (therefore not being connecte with a synchronization line) shall be conntected to its predecessor using a light dotted line. Based on this approach no “orphan notes” appear on the harpsheet.

  }(RS_MG_001)

- [RS_HN_016] **shall support parts** {

  It shall be possible to divide the music in parts. Parts shall be easily identified on the harpnote sheet. Parts may have a name which shall be shown.in the harpnote sheet.

  }(RS_MG_001)

- [RS_HN_017] **denote measure starts** { Measure starts shall be denoted by a small ellipse above the first note/rest of a measure. }(RS_MG_001)

- [RS_HN_018] **use particular line width**{ Lines shall be drawn as follows:

  Notes Thick

  Melody lines Medium

  Secondary Melody line Thin

  Jump lines Thick

  Default Thin

  }(RS_MG_001)