ZupfNoter Requirements

Bernhard Weichel (www.weichel21.de)

18.5.2014

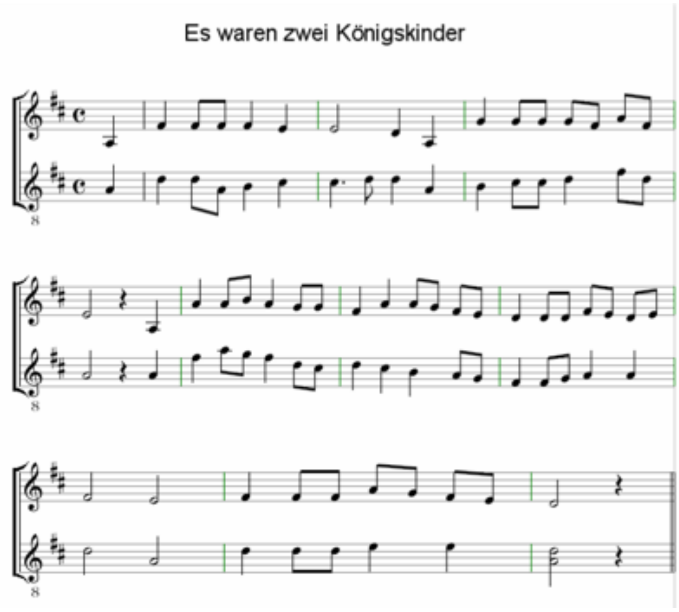
Inhaltsverzeichnis

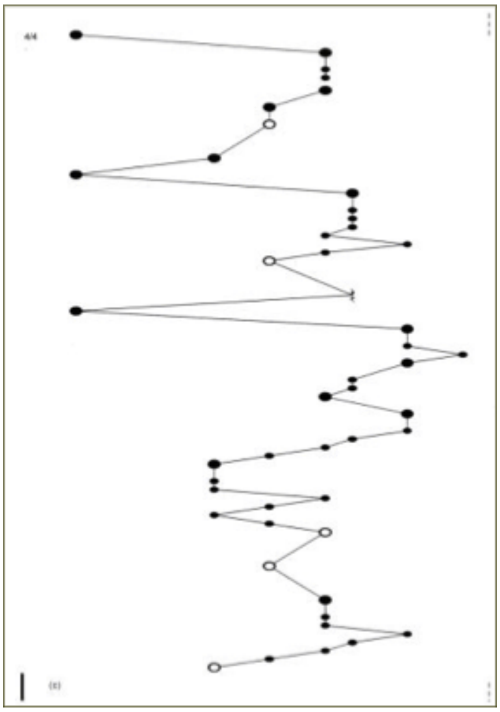
1 Introduction

Zupfnoter is an attempt to implemente a tool to create music sheets for “Table Harp” (http://Table Harpharfen.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 piece





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)

3.2 Input

- [RS_IN_001] **Shall be able to process various input formatc** { 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 curent 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] **Feedback to the entire tune** { The entire tune shall be played upon requrest}(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)

3.3 Output

- [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 paricular to support the fact that private userd 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 scheet is filled.}(RS_MG_002)

3.4 Support of Table Harp notation

- [RS_VN_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_VN_002]**May Support the lyrics**{ Zupfnoter may be able to print the lyrics on the right margin of the page.}(RS_MG_001)
- [RS_VN_003]**Shall 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 tune are connected by a solid line

}(RS_MG_001)

- [RS_VN_004]**Shall Suport chords**{ Zupfnoter shall support notes which are played simultaneously. This is indicated by dotted lines connecting the particular notes of the chord.}(RS_MG_001)
- [RS_VN_005]**Shall repetitions**{ Zupfnoter shall be able to denote repetitions by drawing a rectangular line backwards.}(RS_MG_001)
- [RS_VN_006]**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)