Table of Contents

Cover sheet with Project title, Team members name, Instructor, course deta

34/40

1. Introduction
    1.1. Purpose
        1.1.1. The purpose of this document is to provide a reference during the development of Vivace Music Notation software in regards to the functionality and requirements of said software.

    1.2. Scope

        1.2.1. As a whole, this software will allow a user to notate music on their Windows 8 machine. This software will take advantage of the touchscreens of Windows 8 tablets and phones by allowing users to write music via touch.

        1.2.2. This software will allow the user to create musical notation both at home and on-the-go. It will also fill a void in the software market that arises from the lack of musical notation software on the Windows Metro store.

    1.3. Definitions, Acronyms, and Abbreviations

        1.3.1. Score - A written representation of a musical piece.
        1.3.2. Staff - A five--lined grid used to represent pitch and rhythm
        1.3.3. Grand Staff - A special staff used for keyboard instruments. It is two staves joined by a bracket.
        1.3.4. Definitions pertaining to Rhythm:
            1.3.4.1. Tempo - The speed of the piece. More exactly, the rate at which beats occur per minute
            1.3.4.2. Time Signature - A ratio of beats per measure, and what note qualifies as the beat
                1.3.4.2.1. Possible notes begin with a double whole note, and further notes are 2^-n.
                1.3.4.2.2. Please see reference 1.4.2 for more information on Time Signatures.
            1.3.4.3.
        1.3.5. Definitions pertaining to pitch
            1.3.5.1. Sharp - a symbol reminiscent of #, indicates a note should be raised half a step.
            1.3.5.2. Flat - a symbol reminiscent of a lowercase b, indicates a note should be lowered half a step
            1.3.5.3. Clef - A symbol used to show what pitch belongs to which line and space on the staff.

1.3.5.4. Key Signature - A representation of what sharps or flats exist in piece of music. This can change at any point in a score.

1.3.5.5. Please see reference 1.4.2 for more information on Key Signatures

1.3.5.6. Please see reference 1.4.2 for more information on pitch naming conventions.

1.4. References

1.4.1. Online Music Dictionary: http://dictionary.onmusic.org/

1.4.2. Basic music theory resource: http://www.essential-music-theory.com/

1.5. Overview

1.5.1. This document contains specific information pertaining to the development of the Vivace Music Notation software. Music is a language, and as such the references above (1.4.1, 1.4.2) should be used in conjunction with this document to further the development team's understanding of music terms. The requirements of this document have been organized by software features.

2. Overall Description

2.1. Product perspective

2.1.1. This software combines the power of notation competitors like Finale and Sibelius, with the intuitiveness of touchscreen controls.

2.1.2. System Interfaces

2.1.2.1. The software will need to interface with synthesizers for turning MIDI data into playback audio.

2.1.3. User Interfaces

2.1.3.1. On both Windows phones and tablets, the user will interact with the software via touchscreen.

2.1.4. Hardware Interfaces

2.1.4.1. The app must be run on a Windows 8 device (including Windows Phone 8, and Windows tablets and PC's).

2.2. Product Functions

2.2.1. Read and write MusicXML documents

2.2.2. Save created scores as MusicXML, and open previously created MusicXML files as scores.

2.2.3. Playback scores via MIDI

2.3. User Characteristics

2.3.1. Vivace is intended to be just as powerful as common notation softwares, while offering a much shallower learning curve. Vivace is intended to be used by users with at least rudimentary music knowledge, and as such will not offer any sort of definitions for

terms or basic theory tutorials. Vivace will require a Windows 8 operating system, and will run on either Windows Phone or Windows tablets and PCs.

2.4. Constraints

2.4.1. Hardware limitations

2.4.1.1. The software will require a sound card for playback

2.4.2. Reliability requirements

2.4.2.1. The software is dependent on the user knowing correct music notational standards.

3. Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

3.1.1.1. There will be a home screen for selecting whether to open a score or create a new one.

3.1.1.1.1. Creating a new score will allow the user to select from templates or a blank score.

3.1.1.2. There will be a screen for editing a selected score. This screen will display staves. The user will be able to choose whether to view all staves, or only a given number. This screen will also display toolbars containing tools for editing. Playback will be enabled during this time. → automatically?

3.1.1.3. There will be a view screen to allow users to view the completed score directly from their device (like a print preview). Users will still be able to choose how many staves to view, and playback will still be enabled.

3.1.2. Software Interfaces

3.1.2.1. The software will interface with a synthesizer to translate MIDI data streams into audio.

3.1.2.1.1. Because this software is being developed for Windows, it will use the Microsoft GS Wavetable Synthesizer by default. → Over specification?

3.1.2.1.1.1. Users with extensive knowledge will be able to select a different synthesizer if they so desire.

has this been specified under user characteristics.

3.2. System Features

3.2.1. Score creation and editing — based on instrument list?

3.2.1.1. The software will allow users to notate music via western music notation define

3.2.1.2. Stimulus/response sequence

3.2.1.2.1. The user opens the application

3.2.1.2.2. The software prompts the user to create a new score or open a saved score

3.2.1.2.3. The user chooses to create a new score

3.2.1.2.4. The software prompts the user to select a template

3.2.1.2.5. The user chooses any template

3.2.1.2.6. The software prompts the user for a score title, composer name, arranger/lyricist name, and copyright, as well as initial key signature and time signature.

3.2.1.2.7. The user fills in the appropriate information and the score is created.

3.2.1.2.8. The software generates the score and staves for each instrument.

3.2.1.2.9. The user may now edit the score by entering in notes on the generated staves.

3.2.1.3. Associated functional requirements

3.2.1.3.1. The software will automatically generate measures as the user adds notes.

3.2.1.3.2. Editing the score is defined as the following

3.2.1.3.2.1. Place notes of any value on an instrument's staff

3.2.1.3.2.1.1. Allow multiple voices per staff

3.2.1.3.2.2. Place dynamic markings

3.2.1.3.2.2.1. pp to ff

3.2.1.3.2.3. Place tempo markings

3.2.1.3.2.4. Place changes in key signature

3.2.1.3.2.5. Place changes in time signature

3.2.1.3.2.6. Place changes in Clef

3.2.1.3.2.7. Place articulations

3.2.1.3.2.7.1. slurs, staccato, tenuto, accents, dead notes, bends, slides, tremolo, harmonics

3.2.1.3.2.8. Place ornaments

3.2.1.3.2.8.1. Grace notes, turns, mordents, trills

3.2.1.3.2.9. Place rehearsal numbers or letters

3.2.1.3.2.9.1. Both will be supported, which one is used is the user's choice

3.2.1.3.2.10. Place repeats

3.2.1.3.2.11. Place special endings and codas.

3.2.1.3.2.12. Please see reference 1.4.1 for definitions of terms used here.

3.2.2. Score saving and opening

3.2.2.1. The software will be able to save created scores to MusicXML and open MusicXML files as a score.

3.2.2.2. Stimulus/response sequence

3.2.2.2.1. The user opens the application

3.2.2.2.2. The software prompts the user to create a new score or open a saved score

3.2.2.2.3. The user opens a saved score

3.2.2.2.4. The software displays the score, which can now be edited

3.2.2.2.5. The user edits the score and chooses to save the score

3.2.2.2.6. Conventional file-saving prompts are followed

3.2.2.3. Associated functional requirements

3.2.2.3.1. Opening and saving scores will be accomplished via a file chooser

part of your system or external?

3.2.3. Score playback

3.2.3.1. The software will be able to playback a score being edited

3.2.3.1.1. Stimulus/response sequence

has edited a score

3.2.3.1.1.1. The user is ~~currently editing a~~ score and wants to know how it sounds. The user selects the playback option

3.2.3.1.1.2. The software, beginning from the selected measure, plays through the score.

3.2.3.1.2. Associated functional requirements

3.2.3.1.2.1. The synthesizer used for playback will be selectable by the user

3.3. Performance Requirements

playback?

3.3.1. The system shall save and load files in under 5 seconds.

3.3.2. The system shall playback scores smoothly, with no unintended changes in tempo, and accuracy of pitch.

measurable?

3.4. Design Constraints

this section

Start thinking about

3.4.1. There are no foreseeable design constraints at this time.

3.5. Software System Attributes

3.5.1. Reliability

what if the interface with the synthesizer fail?

3.5.1.1. Due to the self-contained nature of the software, there will be no reason to expect the software to be incapable of running at any given time as long as it meets minimum system requirements and it does not crash due to a bug.

specified?

what?

3.5.1.2. Security

3.5.1.2.1. Due to the self-contained nature of the software, and the nature of its operation, there are no security risks to the user.

3.5.2. Maintainability

Copyright violations?

3.5.2.1. Once completed, the software will only need to be maintained if bugs are found, or new features are to be added.

3.5.3. Portability

3.5.3.1. The system will only be available on a Windows platform. Porting to other platforms is not within scope at this time.