Vivace

Deanna Biesan

Cameron Cordes

Nitin Sharma

Professor Subburaj

Software Engineering I

Table of Contents

1. Introduction
   1. Purpose
   2. Scope
   3. Definitions, Acronyms, and Abbreviations
   4. References
   5. Overview
2. Overall Description
   1. Product Perspective
   2. Product Functions
   3. User Characteristics
   4. Constraints
   5. Assumptions and Dependencies
3. Requirements
   1. External Interface Requirements
   2. System Features
   3. Performance Requirements
   4. Design Constraints
   5. Software System Attributes
4. Introduction
   1. Purpose
      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.
   2. Scope
      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.
      2. This software will allow the user to create musical notation both at home and on-the-go, which will fill a void in the software market that arises from the lack of musical notation software on the Windows Metro store. The current Windows Metro store does not have applications which handle musical score creation by the users and playback through the application of user created scores.
   3. Definitions, Acronyms, and Abbreviations
      1. Score - A written representation of a musical piece.
      2. Staff - A five--lined grid used to represent pitch and rhythm
      3. Grand Staff - A special staff used for keyboard instruments. It is two staves joined by a bracket.
      4. Definitions pertaining to Rhythm:
         1. Tempo - The speed of the piece. More exactly, the rate at which beats occur per minute
         2. Time Signature - A ratio of beats per measure, and what note qualifies as the beat
            1. Possible notes begin with a double whole note, and further notes are 2^-n.
            2. Please see reference 1.4.2 for more information on Time Signatures.
      5. Definitions pertaining to pitch
         1. Sharp - a symbol reminiscent of #, indicates a note should be raised half a step.
         2. Flat - a symbol reminiscent of a lowercase b, indicates a note should be lowered half a step
         3. Clef - A symbol used to show what pitch belongs to which line and space on the staff.
         4. Key Signature - A representation of what sharps or flats exist in piece of music. This can change at any point in a score.
         5. Please see reference 1.4.2 for more information on Key Signatures
         6. Please see reference 1.4.2 for more information on pitch naming conventions.
      6. MIDI – Musical Instrument Digital Interface is a protocol which allows musical instruments and computers to communicate with one another
      7. Synthesizer – computer program used for audio generation
      8. MusicXML – XML-based file format for representing musical notation
      9. Ornaments – musical flourishes
      10. Articulations – marks that affect the transition or continuity on a note or between notes or sounds
   4. References
      1. Online Music Dictionary: <http://dictionary.onmusic.org/>
      2. Basic music theory resource: <http://www.essential-music-theory.com/>
   5. Overview
      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 and are outlined in section 3.
5. Overall Description
   1. Product perspective
      1. This software combines the power of notation competitors like Finale and Sibellius, with the intuitiveness of touchscreen controls.
      2. *System Interfaces*
         1. The software will need to interface with synthesizers for turning MIDI data into playback audio.
      3. *User Interfaces*
         1. On both Windows phones and tablets, the user will interact with the software via touchscreen.
      4. *Hardware Interfaces*
         1. The app must be run on a Windows 8 device (including Windows Phone 8, Windows tablets, and PCs).
   2. Product Functions
      1. Read and write MusicXML documents
      2. Save and edit created scores as MusicXML, and open previously created MusicXML files as scores.
      3. Playback scores via MIDI
   3. User Characteristics
      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.
   4. Constraints
      1. Hardware limitations
         1. The software will require a sound card for playback
      2. Reliability requirements
         1. The software is dependent on the user knowing correct music notational standards.
6. Specific Requirements
   1. External Interface Requirements
      1. User Interfaces
         1. There will be a home screen for selecting whether to open a score or create a new one.
            1. Creating a new score will allow the user to select from templates or a blank score.
         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 automatically during this time.
         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.
      2. Software Interfaces
         1. The software will interface with a synthesizer to translate MIDI data streams into audio.
   2. System Features
      1. Score creation and editing
         1. The software will allow users to notate music via western music notation
         2. Stimulus/response sequence
            1. The user opens the application
            2. The software prompts the user to create a new score or open a saved score
            3. The user chooses to create a new score
            4. The software prompts the user to select a template
            5. The user chooses any template
            6. The software prompts the user for a score title, composer name, arranger/lyricist name, instrument list, and copyright, as well as initial key signature and time signature.
            7. The user fills in the appropriate information and the score is created.
            8. The software generates the score and staves for each instrument
            9. The user may now edit the score by entering in notes on the generated staves.
         3. Associated functional requirements
            1. The software will automatically generates otherwise blank measures as the user adds notes.
            2. Editing the score is defined as the following

Place notes of any value on an instrument’s staff

Allow multiple voices per staff

Place dynamic markings

pp to ff

Place tempo markings

Place changes in key signature

Place changes in time signature

Place changes in Clef

Place articulations

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

Place ornaments

Grace notes, turns, mordents, trills

Place rehearsal numbers or letters

Both will be supported, which one is used is the user’s choice

Place repeats

Place special endings and codas.

* + 1. Score saving and opening
       1. The software will be able to save created scores to MusicXML and open MusicXML files as a score.
       2. Stimulus/response sequence
          1. The user opens the application
          2. The software prompts the user to create a new score or open a saved score
          3. The user opens a saved score
          4. The software displays the score, which can now be edited
          5. The user edits the score and chooses to save the score
          6. Conventional file-saving prompts are followed
       3. Associated functional requirements
          1. Opening and saving scores will be accomplished via a file chooser
    2. Score playback
       1. The software will be able to playback a score being edited
          1. Stimulus/response sequence

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

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

* + - * 1. Associated functional requirements

The synthesizer used for playback will be selectable by the user

* 1. Performance Requirements

* + 1. The system shall save and load files in under 5 seconds.
    2. The system shall playback scores smoothly, with no unintended changes in tempo, and accuracy of pitch.
  1. Design Constraints
     1. There are no foreseeable design constraints at this time.
  2. Software System Attributes
     1. Reliability
        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.
        2. Security
           1. Due to the self-contained nature of the software, and the nature of its operation, there are no security risks to the user.
     2. Maintainability
        1. Once completed, the software will only need to be maintained if bugs are found, or new features are to be added.
     3. Portability
        1. The system will only be available on a Windows platform. Porting to other platforms is not within scope at this time.