Abstract:

Our project is going to be a mobile music notation application for the Android platform of devices. We decided on this project because of the lack of applications that fill this role on the Google Play store as well as the Apple App Store. By the end of the semester we plan to have a functioning app that can take user input in and play it back with rhythmic accuracy.

Introduction:

When it comes to mobile platforms, there is a market for musical notation software. A good notation software would allow musicians to jot down ideas on the go, playback said ideas, and read notation files such as XML, gp5, tux, etc. so that a musician can download files from the internet and view them (as well as play them back) on their mobile device.

The existing problem is no such mobile application exists. Existing applications either suffer from clunky UI or lack of features. We believe there exists room in the marketplace for a notation software app that will supply the best of the existing apps, while trimming away the unnecessary features and improving upon those that already exist.

Solving this problem could open many doors for how musicians share and create music together. As it stands, PC software such as Finale and Sibellius are the most widely used tools for creating sheet music. The speed at which musicians can collaborate on such projects is hindered, due to only being able to create and edit these files from a PC. With a proper mobile app, musicians could very easily quickly create ideas, and share them with their peers for further collaboration on the fly. The app could also serve as a good teaching tool in private settings, and as an excellent practice tool. A musician could simply download a song to their device and, using the playback feature, play along with the music.

Related Work:

A number of paid and free musical notation applications exist in the form of mobile and desktop applications. The approaches that these software systems take to solve the problem of musical notation differ in a variety of ways, chief among them: forms of user interaction with the applications, features of the applications, whether the music written can be played back, and the format in which the application exports the music. Some of the applications presented the users with a list of notes that could be dragged onto the screen while others just had the user touch and scroll through to find the correct note types. In addition, some allowed the written music to be played back by the user after an instrument has been selected, while others do not include this feature. Finally, the export formats of these applications vary among musicXML, midi, png, jpg, and pdf.

Methodology:

To develop this application we plan to break down the project into sections to put together to make the functioning application. The User Interface will probably be the first thing we work on to lay the foundation for all the functionality that we decide to build into the application. After the UI is in place then we can write the code to allow for the insertion and deletion of notes with their duration and pitch easily changeable. After we get the basic functionality in place we would like to try to incorporate MIDI, which is a type on input interface where you can plug in a keyboard and play the notes as the software inputs the notes making it easier and quicker to write music.

References:

Music Notation Editors and Included Features:

1. Maestro –includes pitch pipe and metronomes; currently in beta testing
2. Ensemble Composer –uses midi for sound; exports midi files
3. Notateme Now –enter music notation via touch; allows the user to take pictures of sheet music with a camera; can be played back
4. Music Composition – exports to printable image file and playable audio file
5. Music Score Pad – offers free version and plus version; drag and drop notation; allows the user to take pictures of sheet music with the camera and then edit the sheet music from the image

Project Milestones:

1. App can display a musical staff
    a. Clefs can be placed on the staff
    b. Key Signatures can be placed on the staff
    c. Time Signatures can be placed on the staff
2. App can accept basic notation input
    a. App now has a basic functioning flow (Splash Screen or Menu Screen takes you to notation screen)
    b. Notes can be placed on staff by touching the staff
    c. Note heads from a breve note to a 64th note are supported
    d. Measures are created when notes are placed with respect to the Time Signature
    e. Note heads can have accidentals placed on them
3. App is capable of saving created files and opening previously created files
4. App can accept advanced notation input

   a. App can now display multiple staffs as well as grand staffs (conjoined staves).
   b. Note heads can have articulations
      i. Slurs, tenuto, accent, and staccato.
   c. Tempo marking dictionary
5. App is capable of MIDI playback
   a. Related staves are played back simultaneously.
   b. Soundfonts are selectable for each staff.
6. App UI facilitates easy reading while playing
   a. App will have options to adjust zoom
   b. App will slowly scroll along while music is played back
7. App can open commonly used file types
   a. Music XML is the priority
      i. Finale, Sibellius, Guitar Pro 5, if possible.