Master Copy and Arrangements

Separating music from layout information

Contact Details
Name: Peter Jonas
Email: pjonas56@gmail.com
Mobile: +44 7507 547 924
Location: Scotland, UK

Online Presence
MuseScore.org (shoogle)
GitHub (shoogle)
SourceForge (shoogle)*
Google Play AppStore (shoogle)

**Please note: my contributions to the C++ project on SourceForge were not in C++ or Qt. See my Bio for details.*

Synopsis

MuseScore already has a facility to extract individual parts from a score, and both versions are updated automatically to reflect changes made to the other copy. At a simple level, an arrangement can be considered to be made up of multiple parts. For example, an orchestral score might contain 20 or more parts, but the same piece arranged for a string quartet would only have 4 parts. It would be advantageous to be able to store the quartet arrangement in the same file as the full version so that changes in one can be reflected in the other. Even if the 4 parts are not exactly the same as the equivalent parts in the full orchestral score, it would still be useful to see them side-by-side. This project would see the creation of the concept of a "Master Copy" in which *all* parts from *all* arrangements are displayed, and "Arrangement Copies" in which only the parts relevant to a particular arrangement are displayed. This idea can easily be extended to include different versions of song lyrics. For example, it would be possible to store multiple lyric translations in one file (i.e. the "Master Copy") and allow the user to pick which is displayed in a given arrangement, or have it chosen for them automatically based on their system language settings. I proposed a similar feature a while ago on MuseScore.org at node/71096#comment-327816 (and again at node/94021#comment-419661) but I have changed some aspects since then.

Benefits

This project would seek to establish firm definitions for the following terms which are often used interchangeably: "part", "instrument" and "staff". This, combined with the other changes explained in the details section, would solve numerous open issues and feature requests:

- ideas-2016#extract-part-from-voices - Extract a different part from each voice
- node/42306, node/18221, node/13584 - Control each voice separately in Mixer
- node/42301 - Change instrument back-and-forth between two (or more) instruments

In fact, not only would this project solve those issues, it would do so in a more robust way than the solutions suggested in those issues. For example, it is common practice for each voice in a staff to represent a different part (e.g. Tenor and Bass), but different voices do not always represent a different part so allowing separate mixer control for voices, or voice extraction, would not always be desirable. This project would solve the underlying issue by allowing the user to define each part, so that the computer "knows" whether or not two voices belong to the same part.

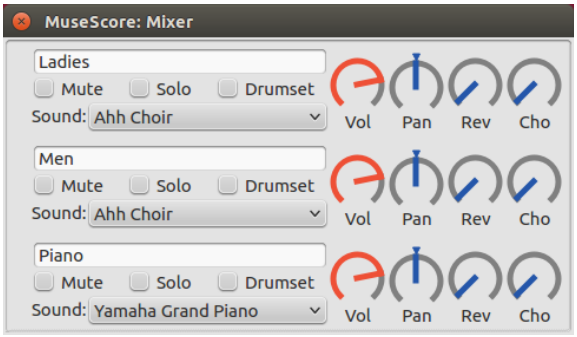
This project would also make the process of collaborative score editing significantly easier. For example, someone (perhaps MuseScore!) could begin the task of creating a definitive free copy of some iconic public domain pieces, such as Handel's "Hallelujah Chorus", in the MSCZ format and invite collaborators to join in with the transcription. Once the orchestral copy was in place a piano reduction would be needed for choirs to rehearse from. Of course, the vocal lines would be identical in the piano vocal score as the orchestral score. By maintaining the piano arrangement in the same file as the orchestral arrangement, any mistakes in the vocal lines could be corrected instantly for both arrangements. Also, collaborators could translate the lyrics into their own language, with all translations stored in the one file. The original lyrics are English, so if an English-speaking user were to download the file then only the English lyrics would be displayed. If an Italian user downloaded the score then the Italian lyrics would be displayed beneath the original English lyrics, which would be shown in italics.

Deliverables

There would be a dropdown list of arrangements, with the default options being "Master" and "Create new arrangement". A score that contains only a Master and no arrangements would essentially be the same as a current MSCZ file, so it would be the default and most users would just continue using the program as they currently do without noticing any change. However, if a user clicks the "create new arrangement" option, they would be presented with a dialog enabling them to give the arrangement a name, and to choose which instruments from Master should be included in the new arrangement, along with more advanced options such as whether multiple instruments should share a staff (e.g. Tenor and Bass), and whether multiple instruments should be grouped into a single part (e.g. create a part called "Bass" which includes both 1st Bass and 2nd Bass). While in Master, the Mixer would display every instrument separately (as it does now), but in an arrangement the Mixer controls would be grouped logically by part (e.g. Bass), with the option to expand parts for separate control of the instruments within the part (1st Bass and 2nd Bass). As previously mentioned, if multiple lyric translations exist then the original language is displayed, along with the user's own language, if it is different to the original (in which case the original is shown in italics). There would also be an option to manually specify which languages are displayed.

Current Mixer:

Currently the user has to choose between saving space in the score by combining multiple parts on one staff, or retaining independent mixer control. Due to the part/staff equivalence, it is not currently possible to have both independent mixer control and multiple parts on one staff.

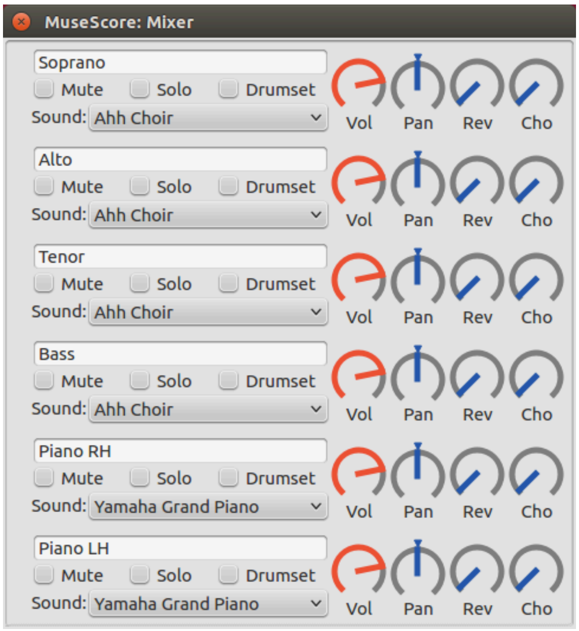


MuseScore: Mixer

Ladies
☐ Mute ☐ Solo ☐ Drumset
Sound: Ahh Choir

Men
☐ Mute ☐ Solo ☐ Drumset
Sound: Ahh Choir

Piano
☐ Mute ☐ Solo ☐ Drumset
Sound: Yamaha Grand Piano



MuseScore: Mixer

Soprano
☐ Mute ☐ Solo ☐ Drumset
Sound: Ahh Choir

Alto
☐ Mute ☐ Solo ☐ Drumset
Sound: Ahh Choir

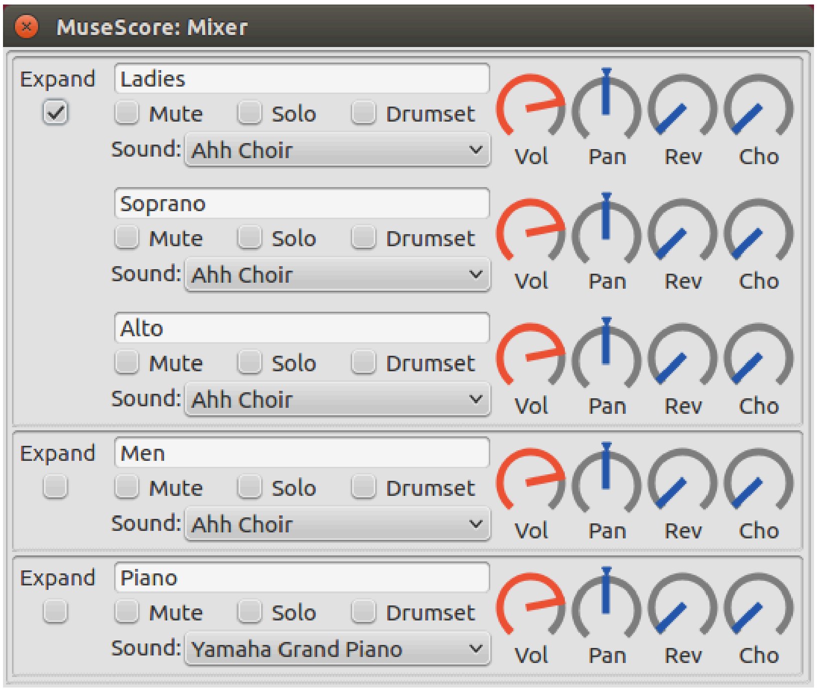
Tenor
☐ Mute ☐ Solo ☐ Drumset
Sound: Ahh Choir

Bass
☐ Mute ☐ Solo ☐ Drumset
Sound: Ahh Choir

Piano RH
☐ Mute ☐ Solo ☐ Drumset
Sound: Yamaha Grand Piano

Piano LH
☐ Mute ☐ Solo ☐ Drumset
Sound: Yamaha Grand Piano

Proposed Mixer layout for Arrangements:



MuseScore: Mixer

Expand ☑ Ladies
☐ Mute ☐ Solo ☐ Drumset
Sound: Ahh Choir

Soprano
☐ Mute ☐ Solo ☐ Drumset
Sound: Ahh Choir

Alto
☐ Mute ☐ Solo ☐ Drumset
Sound: Ahh Choir

Expand ☐ Men
☐ Mute ☐ Solo ☐ Drumset
Sound: Ahh Choir

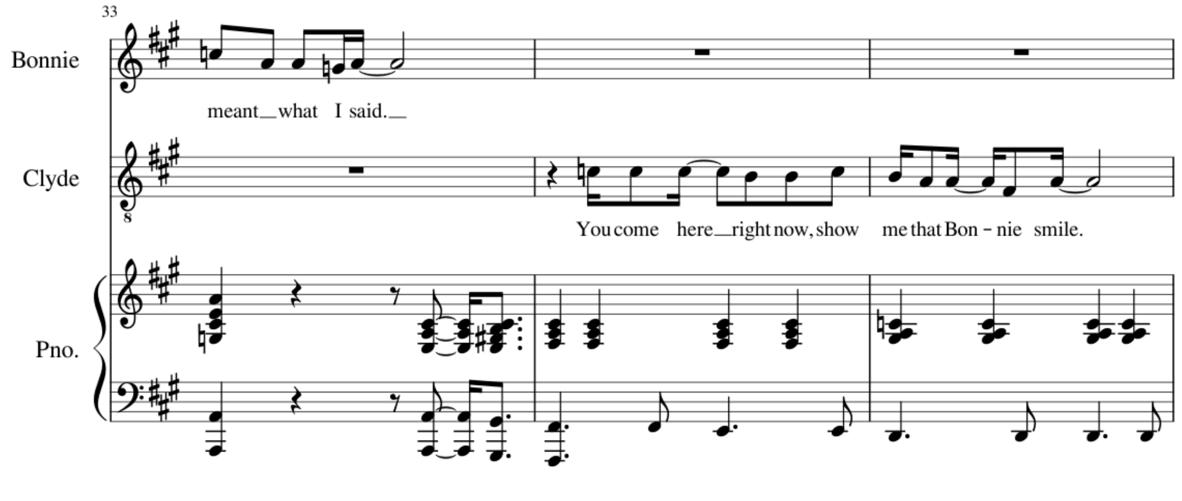
Expand ☐ Piano
☐ Mute ☐ Solo ☐ Drumset
Sound: Yamaha Grand Piano

Within an arrangement multiple parts are allowed to share a staff, but they can still be controlled independently in the Mixed because the instruments were defined separately in the Master copy. By asking the user to define a part, it is possible to group the instruments in a logical way in the Mixer.

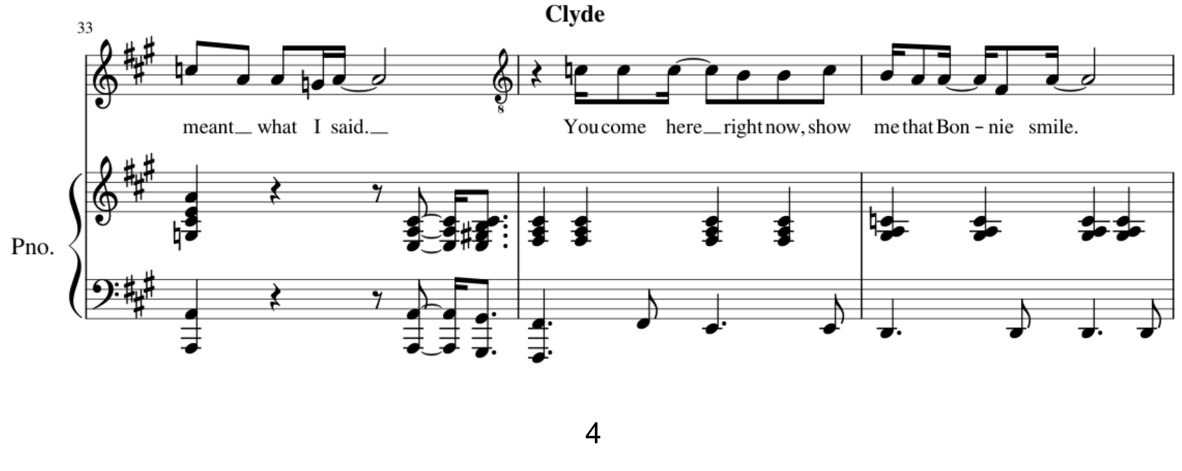
Project details

For the sake of less advanced users, Master would behave just like MuseScore behaves now. However, advanced users would be encouraged to enforce a strict *"one instrument per staff"* rule in Master. (Some instruments - e.g. a piano - can occupy multiple staves, but a staff cannot have multiple instruments.) The term "part" is only used in arrangements, where it could refer to one or more instruments. The arrangements would be stored in the MSCZ file along with Master. Initially, arrangements would just have a name, a number of staves, a list of parts (where each part is a list of instruments from Master), and a list of which parts appear on which staff. At first there would be no note or layout information stored in the arrangement as this would be carried over from Master, but layout information could be added at some point in the future, or perhaps later on in the project if there is time. However, it is likely that detailed manual layout adjustment would no longer be as important given that users would now have the ability to put multiple parts on a single staff. Therefore, the main focus of any time left at the end of the project would be on improving MuseScore's automatic layout algorithms. The new definitions of "part" and "instrument", and their separation from the definition of a "staff" create plenty of opportunity for doing this.

Current layout:



Potential improvement:



In this example, the two vocalists sing on the same system, but not at the same time so they could be combined onto one staff. The existing “hide empty staves” feature cannot do this because it can only hide measures if the staff is empty for the whole system. Though it is possible to manually put the vocalists on the same staff in the current version of MuseScore, doing so removes the ability to control them separately via the Mixer. This would not be the case if the proposed improvements were made to define a “part” separately to a “staff”.

This is just one area of improvement among many that would become possible thanks to the new definitions of “part”, “instrument” and “staff”. Of course, the improved layout would not be achieved by the new definitions alone; the layout algorithms would also need to be changed. Large changes to the layout algorithms may not be achievable in the time available, but the mere existence of arrangements would encourage a new way of thinking among the developers that could see the layout improvements made at a later date.

Project Schedule

The project would count towards my Masters degree and would be the topic of my dissertation. The deadline for final submission of the dissertation essay is still to be confirmed by my university, but it will probably be in the final week of August - i.e. around the same date as the final submission deadline for GSoC - so ideally all coding would have been completed a week or so before this. I will also have a supervisor within the University with whom I shall have regular face-to-face meetings to check up on progress.

The target is to have a basic implementation of arrangements completed by the halfway point, along with the new Mixer. A basic implementation could consist of one-way editing, i.e. edits in Master appear in the arrangements, but the arrangements themselves cannot be edited. This should be realistically achievable in the time available, because it would be equivalent to automating the process of creating a new score and copying the notes from the staves in Master into different voices in the staves in the arrangement. The remaining time would be used to give more control over arrangements and improve the automatic layout algorithms.

Bio

I have already contributed to MuseScore’s code and documentation, and I am an active member of the community forums where I frequently answer questions from new users. I use the alias “shoogle” on MuseScore.org, and have been mainly responsible for packaging on Linux via the Ubuntu PPA and the new AppImage format. I have contributed a little to MuseScore’s main C++/Qt code, and I would like to do more of this in the future.

I am currently studying for a Masters in Advanced Computer Science at the University of St Andrews. My undergraduate degree was in Physics, from which I learned much about sound and signal processing as well as computer modelling and coding. I put the knowledge to good use a couple of years ago when I participated in a research survey with the Marine Biology department. The survey used underwater acoustic equipment on a drifting buoy to triangulate the positions of harbour porpoises based on the characteristic clicking noises that they use to communicate. My task was to create a wireless data link to allow the buoy's acoustic equipment to be monitored remotely from the research yacht, which I achieved using a Raspberry Pi and shell scripts. I also undertook a computational project in laser physics. The project used a conjugate gradient optimisation routine - written in Fortran - to calculate phase patterns for a spatial light modulator (SLM). The SLM is used to produce optical holograms by causing interference in a reflected laser beam. The shape of the hologram (i.e. its intensity profile) is determined by the input phase pattern, which must be calculated to obtain a hologram that would be useful for atom trapping experiments. I also produced a poster explaining my work on the project, which is available here.

I have always been adept with technology, and during my undergraduate years I gained such a passion for programming that I decided to stay another year to gain a formal qualification in the subject. Modules I have done on my Masters course include Human Computer Interaction, Constraint Programming, Multicore Programming in Haskell, and Advanced Networks. I have gained experience programming for a number of different devices, including Android Tablets and Microsoft Kinect. Although I have not had much experience with C++ or Qt, I have a good deal of experience with Java and C#, and I have done an entire module dedicated to object oriented programming.

As a child I received a basic musical training on the trumpet and violin, but in my time at university I discovered that singing is my true passion. I have been a member of various choirs and performing groups. In particular, I received a vocal scholarship with St Andrews Chorus - the largest choir of its kind in Scotland - and was elected Vice-President of the St Andrews Gilbert and Sullivan Society. I am considered the technology guru in both groups, and introduced MuseScore to the members of each. This experience interacting with the target audience, together with my background in Physics and Computer Science, should make me ideally suited to undertaking a project with MuseScore.

Thank you for your consideration. Please see my CV/resume for more details, and do not hesitate to contact me if you require any additional information.