FOSSProF Final Report

VisualScore

Max Eidinoff

Project Overview

Project Title: VisualScore

Project Summary: VisualScore is a new open-source software allowing composers to digitally create sheet music containing avant-garde notations. This program is built on top of the neoscore Python library which already renders most of the symbols that VisualScore needs. The goal of this project was to build a GUI to interact with 2–3 core neoscore features.

Target Audience: Avant-garde composers who are not proficient in Python, particularly those in academia. Secondary groups could potentially be publishing companies with in-house engraving services and universities.

Code Repository: https://github.com/VisualScore/VisualScore

Project Activities and Progress

Work Completed: We originally set out to implement a GUI for 2–3 neoscore features. After partially implementing the first feature, we encountered bugs requiring us to instead shifted our focus to gain a better understanding of how neoscore and its major dependency, PyQt5, interact. To that end, we created a public Notion webpage (linked in VisualScore's README file) which outlines both Qt and neoscore's file structure with comments indicating the most relevant files and potential approaches for VisualScore's next steps in development. There are also links to the most important Qt documentation pages and YouTube tutorials for VisualScore. Through this research and resource collection, the groundwork is now in place to reduce onboarding time for future developers as well as create a clearer path forward to attain a first prototype of VisualScore.

Technical Milestones: A GUI has now been set up to render the neoscore app, add and remove pages, and add and move many SMuFL library music symbols through a drag and drop interface. A public Notion webpage has also been published with resources to speed up future developer onboarding.

Challenges and Solutions: The biggest challenge we faced was finding student developers who were both qualified and interested in working on an arts open-source software project. After sending a job posting to the entire computer science department through a professor, I received

resumes from 6 interested students. Of those, 3 of which appeared to be qualified and I hired them. For the next month, they began work while I attempted to sort out their payment through HR. Since I am a student and there is no faculty supervisor for this project, I was prevented from accessing grant funds for just over a month while I was redirected between several offices. This continued until it was decided to instead make this project a job posting on SMILE through the library HR and OSPO. Thankfully the developer team was very patient during this period of unpaid work.

However, over the next several weeks after payment was set up, it became clear that this project ultimately demanded more time than 2 of the developers were able to sustain. By this time it was already June and there was only 1 developer left working on the project. Given the amount of time that had been spent onboarding developers and the fast-approaching FOSSProF deadline, I decided that it would be better to shift project focus rather than attempt to hire new developers. From that point on, the last developer primarily focused on research to create the Notion resource page. As a result, future hiring can be done faster to avoid this problem.

Outcomes and Impact

Project Impact: Before beginning this project, I surveyed 86 composers about their experience using their current music notation software. Across 4 different software options, the single biggest complaint was a lack of features supporting contemporary music notation. VisualScore aims to relieve this pain point. Through this goal, the project has the potential to become the only software that truly serves avant-garde composers. However, there is more work to be done to reach an MVP prototype and achieve this positive impact.

Community Engagement: We have determined that it would be more productive to put resources into engaging with the open-source community after we have a more complete prototype. That way there is more direction laid out for the project before the community continues its development through smaller changes. We also plan on engaging with the composition community at that point for initial user testing and feedback. This feedback will inform VisualScore's future direction.

Sustainability / Future Plans: Now that we have the beginnings of a prototype, we are planning to use this progress (possible due to the FOSSProF) to advocate for future grant funding from other sources. This continued fundraising will allow us to build out a new project team to finish the initial prototype. Then we will conduct user testing with composers and generate interest within the open-source community. From there we will evaluate what future steps should be taken at that point.

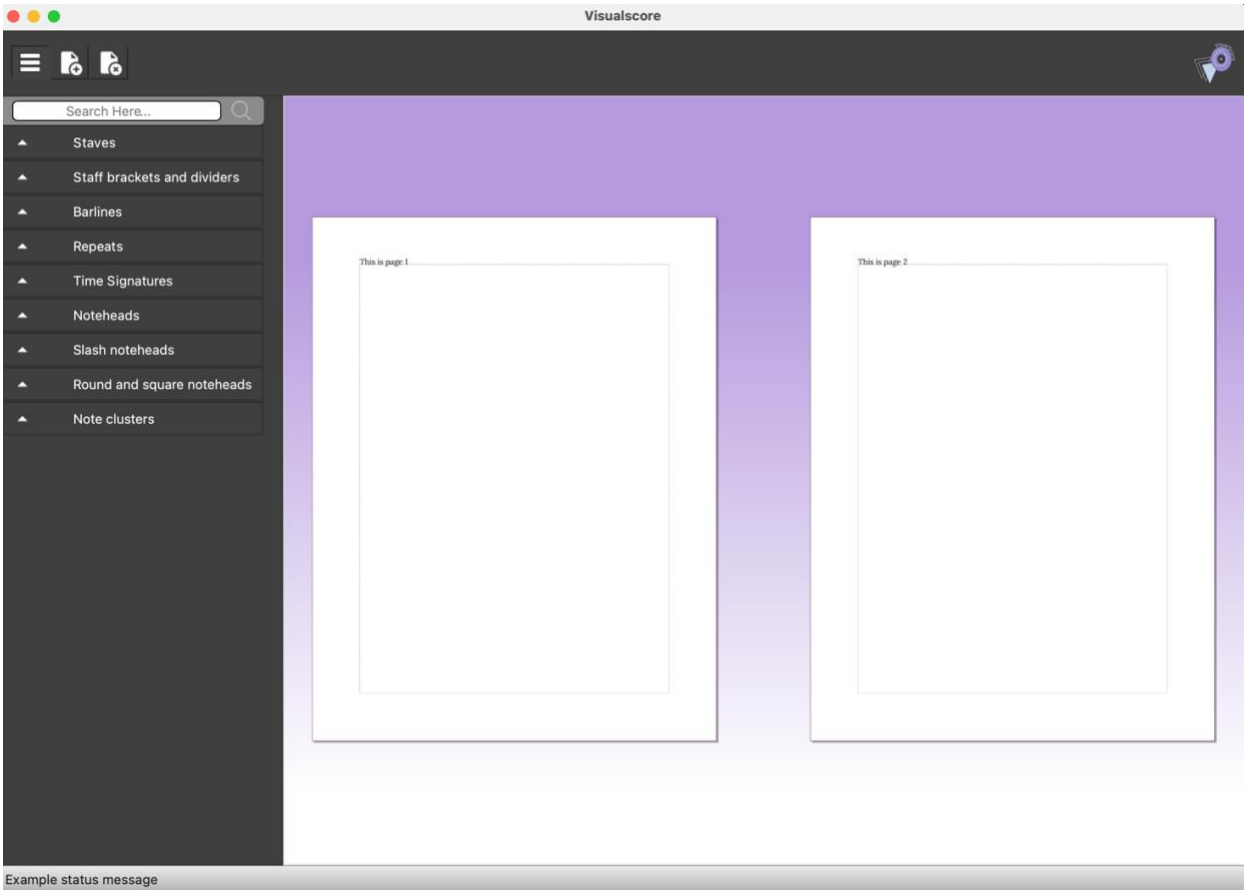
Lessons Learned: The biggest takeaway from this project is that I will need to build an administrative team in addition to the development team in order to assist with organization and create a collaborative decision-making process. Especially since I lack technical expertise, I need additional perspectives to more effectively progress this project. Being the sole decision maker has made it more difficult to foresee the need to pivot. Administrative collaborators would have

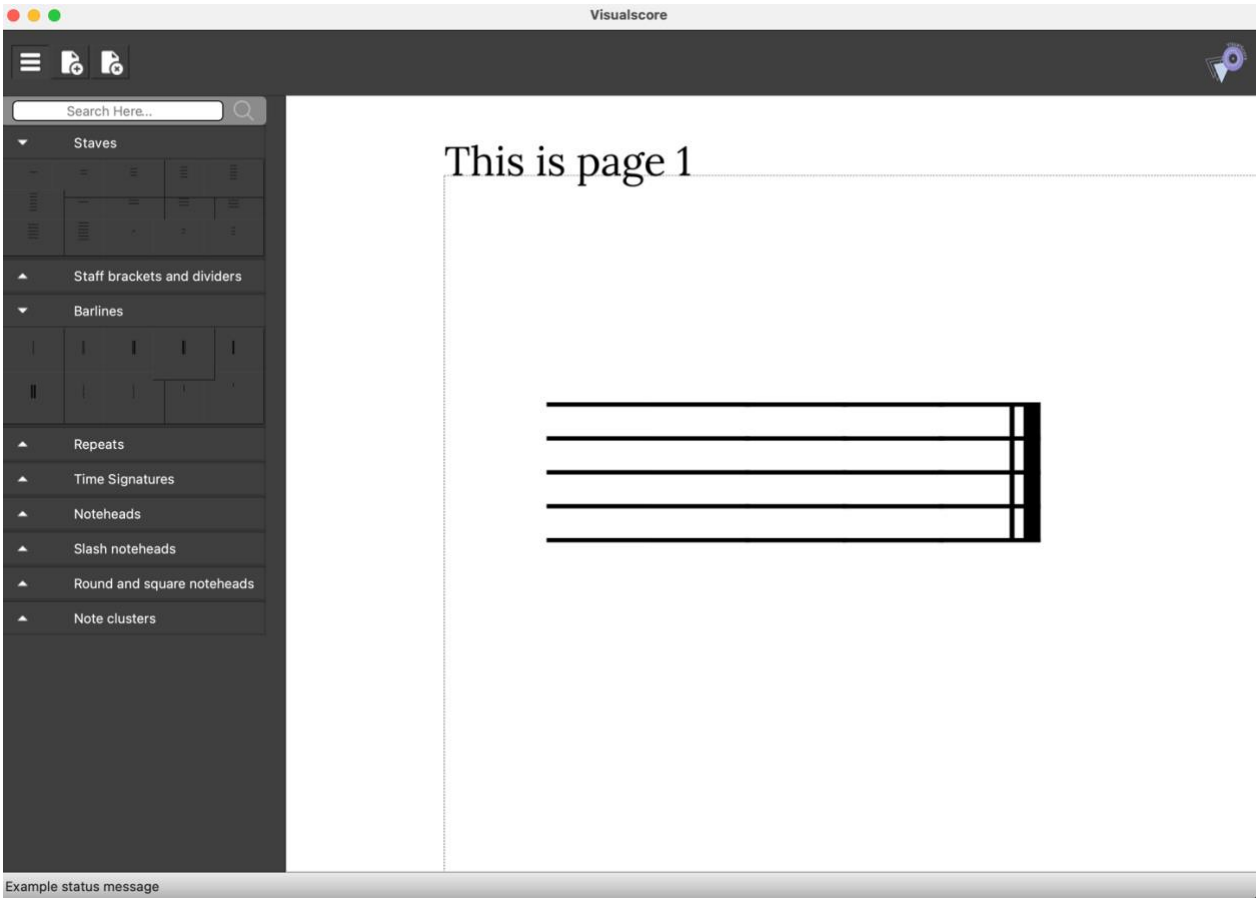
likely made it easier to shift directions earlier during the grant period or even speed up the hiring process.

Attachments:

Notion webpage: https://yielding-bucket-c99.notion.site/QT-PyQt5-Neoscore-Documentation-for-Visualscore-e6e2957dbcc24643a1bf6e7c706a9f75?pvs=4

Screenshots from prototype:




Visualscore
Search Here...
Staves
Staff brackets and dividers
Barlines
Repeats
Time Signatures
Noteheads
Slash noteheads
Round and square noteheads
Note clusters
This is page 1
Example status message