Impromptune

Sprint 1 Retrospective

Team 5
Ben Ahlbrand
Chris Doak
Sean Phillips
Jacob Richwine

1. What went well?

We consider Sprint 1 a major success and nearly completed all of the intended user stories. We only missed a few tasks and those will be incorporated into the beginning of Sprint 2.

- Loading of Music.XML files into GUI Renderer
- Allow manual composition from visual keyboard
- Allow selection of manual composition options
- Show composition as score in GUI, and update when notes are played
- Allow playback of composition, and hear piano note played
- Allowed printing to PDF of current loaded score
- Worked well as a team, and multiple people worked together to complete tasks
- Combined four different projects(Zong! renderer, Zong! player, visual piano, and GUI) into one project, and made it stable
- Our projected hourly tasks estimates were close to actual time taken
- Accomplished most of what our vision was before the sprint, and had to make no major issues or underlying architecture changes

User Story Progress:

As a user, I would like to see the composed music as sheet music. (100% Complete)

As a user, I would like to import a music file to load into the program. (100% Complete)

As a user, I would like to save and load my work. (75% Complete)

As a user, I would like to choose from various instruments. (100% Complete)

As a user, I would like to play each channel individually or concurrently. (Removed)

As a user, I would like to manually compose music. (100% Complete)

As a user, I would like to be able to compose via a GUI. (80% Complete)

As a user, I would like to be able to edit / interact with the composition (80% Complete)

2. What did not go well?

A lot of time spent this sprint was dealing with IntelliJ, github, and project file management issues. At first we were pushing files to github that were unique to our computers and they were wiping out everyone else's workspace configuration. We also had many issues with the custom libraries Zong uses and had to rebuild the dependencies manually many times. Several times we would merge and when someone else pulled, it broke their project. Since Zong was built with the Eclipse IDE, we had to import it into intellij and one of the libraries they used (Lombok) which is also a plugin is not fully functional for intelliJ.

As far as workflow goes, we had an issue where members were waiting for other members to complete their tasks before they could implement their own code. This caused a hang up as team members couldn't proceed without other tasks being finished first. This caused multiple people to work on the same code at same time and have conflicts over which one to keep.

We did not finish parts of three user stories. The GUI was not finished, the undo/redo functionality was not implemented, and the save composition to MusicXML format was not completed.

The task of undo/redo took longer than expected, as gauging from the Zong documentation, it appeared to be integrated functionality however, as of the latest release the library's implementation is missing necessary components to utilize the CommandListener classes as intended. Modifying the current project code or implementing our own method was more complicated than projected.

The option to save as MusicXML was not completed due to the fact that there is not a current way to parse the Zong score object to XML. Creating a parser of this magnitude is significant work and we plan on utilize the JFugue to MusicXML class in Sprint 2.

3. How should you improve?

Although we had a very successful Sprint 1, there are a few things we can improve upon. We can improve Sprint 2 by improving the ordering efficiency of tasks by making an effort to consider what pieces other team members need, and when there is a hang up in the workflow, work on another aspect if possible or offer/ask for help to speed up the bottleneck task.

We have added optimizing and fixing intelliJ/Github issues to the Spring 2 backlog to alleviate time spent fixing project build issues.

While Sprint 1 had good estimates on tasks, for Sprint 2, we should break the tasks into smaller chunks and divide them up more.