**Impromptune**

**Sprint 1**

**Team 5**

Ben Ahlbrand

Chris Doak

Sean Phillips

Jacob Richwine

# 1. Sprint Overview

This sprint will primarily concern itself with UI layout / rendering and file access. While lacking in many features and utility, by the end of the sprint we plan on having the GUI and GUI controls in place and ready to utilize for future sprints.

Completed user functionality by end of sprint include:

* Usable GUI and controls
* Open MusicXML files
* Save MusicXML files
* Display parsed MusicXML in graphical renderer
* Display Interactive Piano
* Allow mouse selection of Piano keys
* Allow selection of manual input options (note type, etc.)
* Allow playback of loaded MusicXML file
* Handle input from Piano and input options

**Scrum Master:** Jacob Richwine

**Meeting Schedule:** We will meet at least once a week on Tuesday/Thursday from 3 to 5.

**Risks/Challenges:**

* Having a coordinated start to the project
  + This project contains vital interactions between objects from different 3rd party code sources and we must be certain that we are all on the same page
  + The first sprint entails creating the GUI and we must have a solid and familiar basis for developing, interacting, and receiving data from the user interface controls
* Implementing / Importing existing libraries into our own project
  + 3rd party code can sometimes be daunting if not maintained or commented well and using the code in a different project may lead to uncertainties and implementation problems
* Display Renderer
  + The majority of the program’s utility relies on the renderer and it must be efficient, stable, and easy to manipulate from different controls

# 2. Current Sprint Detail

|  |  |  |
| --- | --- | --- |
| As a user, I would like to be able to compose via a GUI. | Estimated Time | Responsible |
| Create Master GUI Frame | 3 Hours | Chris |
| Divide Master GUI Frames | 2 Hours | Chris |
| Create Manual Input Frame | 5 Hours | Chris |
| Create Playback Frame | 2 Hours | Chris |
| Create Generation Frame | 2 Hours | Chris |
| Create Dropdown Menus | 1 Hour | Chris |
| Apply CSS to GUI | 5 Hours | Chris |
| Implement Logic for GUI controls | 7 Hours | Chris |
| Create a view frame for the Zong! renderer | 3 Hours | Chris |
| Design a GUI Layout using JavaFX Builder | 3 Hours | Chris |

|  |  |  |
| --- | --- | --- |
| As a user, I would like to manually compose music. | Estimated Time | Responsible |
| Implement GUI for visual keyboard | 5 Hours | Sean |
| Implement key press logic for keyboard | 2 Hours | Sean |
| Allow selection of composing options | 4 Hours | Sean |
| Handle selected options and keyboard and send to controller | 8 Hours | Sean |

|  |  |  |
| --- | --- | --- |
| As a user, I would like to play each channel individually or concurrently. | Estimated Time | Responsible |
| Implement playback frame logic | 5 Hours | Sean |

|  |  |  |
| --- | --- | --- |
| As a user, I would like to choose from various instruments. | Estimated Time | Responsible |
| Choose Instrument for playback | 3 Hours | Sean |
| Choose Instrument for editing | 3 Hours | Sean |

|  |  |  |
| --- | --- | --- |
| As a user, I would like to save and load my work. | Estimated Time | Responsible |
| Add File-Open Logic | 1 Hour | Ben |
| Add File-Save Logic | 2 Hours | Ben |
| Implement file handling code | 4 Hours | Ben |

|  |  |  |  |
| --- | --- | --- | --- |
| As a user, I would like to import a music file to load into the program. | Estimated Time | Responsible | |
| Implement Logic for handling MusicXML | 3 Hours | Ben |
| Send parsed MusicXML to renderer interface | 2 Hours | Ben |

|  |  |  |
| --- | --- | --- |
| As a user, I would like to see the composed music as sheet music. | Estimated Time | Responsible |
| Inspect current Zong! renderer code base | 2 Hours | Jacob |
| Transfer needed parts of Zong! into code base | 5 Hours | Jacob |
| Adapt Zong! code to current project | 5 Hours | Jacob |
| Implement Interface for updating the layout | 5 Hours | Jacob |
| Embed the renderer output into overall GUI | 5 Hours | Jacob |

|  |  |  |
| --- | --- | --- |
| As a user, I would like to be able to edit / interact with the composition. | Estimated Time | Responsible |
| Implement Undo/Redo | 5 Hours | Ben & Jacob |
| Extend Zong! code to allow cursor position for input | 8 Hours | Ben & Jacob |
| Inspect Zong! code for project modifications | 8 Hours | Ben & Jacob |

# 3. Backlog

**Functional:**

1. As a user, I would like to choose from different parameters for algorithmic composition.
2. As a user, I would like to be able to re-generate pieces of previously generated patterns.
3. As a user, I would like to add dynamics to my music.
4. As a user, I would like to be able to use a MIDI controller for manual input (if time allows).
5. As a user, I would like to have control over the generative patterns’ parameters.
6. As a user, I would like to print my music.
7. As a user, I would like to manipulate the musical renderer to allow for easier modification
8. As a developer, I would like the music generation process to play live feedback when a user provides input via a MIDI device (If time allows).
9. As a developer, I would like to easily add new musical generation templates (if time allows).
10. As a developer, I would like to easily create new styles of music (if time allows).
11. As a developer, I would like to create a new file format to be used by the program.
12. As a developer, I would like to have the music generation process take a maximum of a few seconds.

**Non-Functional:**

1. We must be able to run this software on all operating systems.
2. The interface needs to be intuitive and allow experienced musicians sufficient control (simple suggestions that can be easily edited, to the note)
3. The GUI must be responsive.
4. The application has a place to display current processing tasks (if time allows).