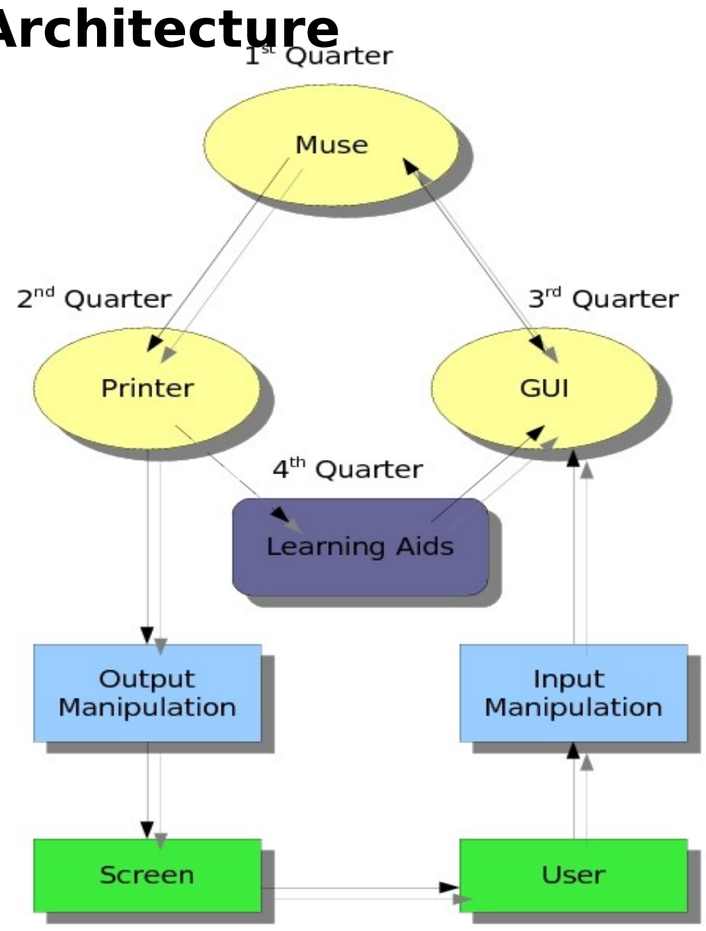
Compositional Software Development
Patrick Mutchler
2006-2007

Abstract
Current music editing software is expensive, complicated, and can be a poor teaching tool. A new, free software designed for amateur composers and music students requires a less powerful editing system and can incorporate learning tools to aid teachers in teaching music theory to students.

Architecture



Version System
The development system for this program has four major release versions, one for each quarter.

0.x – File input, ASCII output
1.x – File input, Graphics output
2.x – GUI input, Graphics output
3.x – GUI input, Graphics output, compositional tools, teaching aids

Expected Outcomes
A usable, free, intuitive program that can create and print simple to mildly complex scores in a variety of ways.

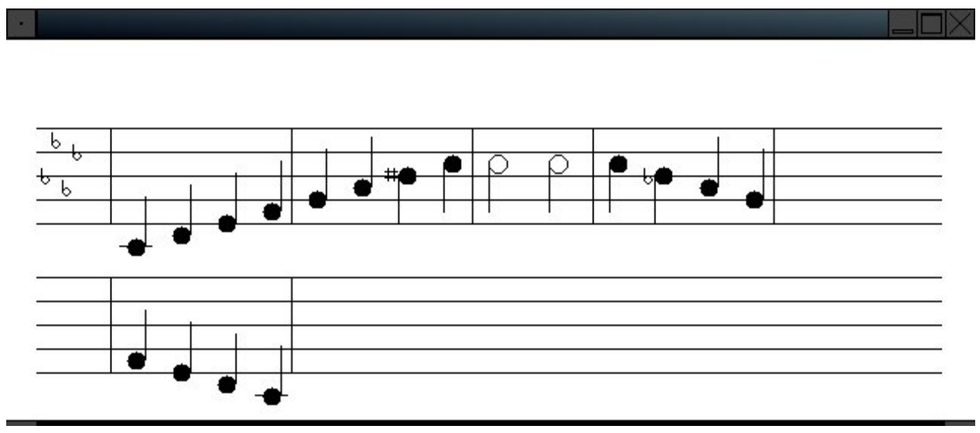
The program will be tested by other students in the classroom to make sure that it meets expectations in terms of usability and simplicity.

Background
Existing editing software falls into two major categories: compositional software and copying software with a focus on visual appeal.

Finale and Sibelius are two examples of compositional software. These programs are aids to composers to make writing music easier and nothing more. I have used some of the Sibelius interface as a model, but without so many complexities.

Lilypond is a program that focuses on music copying and making computer printed scores as appealing as hand written scores. Lilypond, however, cannot be used to create or edit scores quickly or easily due to a complicated file input system. Lilypond does not relate to my software at all, but is an example of software being developed in this field for other purposes.

Sample 1.x Output



Methods Used
getMusic – Reads file input
paint – Calls the drawing methods
parseKeySig – Converts file key signature input into a usable form
printNote – prints a note based on pitch and duration
printMeasure – prints the notes in a measure determined by an array
printScore – prints the measures in a score determined by a matrix