Heqet Application Report

By Isaac Reilly

advised by Donya Quick

continuation of my senior project for the Yale University major in Computer
Science

spring 2016

1   Overview

Last semester, I created the Heqet library to improve music representation and manipulation in Haskell and to export Lilypond code to make pleasing printed scores. My project this semester was to create a graphical application using the Heqet library, with some additional features like multiple selections and a automatically updating dependency graph.

2   How did it turn out?

The project turned out to be vastly more than I could finish in a semester, but my efforts have laid a solid foundation for future work. The graphical interface, which I built using the `threepenny-gui` library, is almost complete and includes navigation, music rendering, and many palettes of buttons. Rendering the music in real time turned out to be an issue: I was hoping to use an speed-optimized form of Lilypond, but this proved impossible. Instead, I created a fast music drawing system which avoided many aesthetic adjustments of the output which slow down Lilypond, such as varying the horizontal spacing of notes.

I adapted the Heqet library for the application and converted it to use the Stack build system for Haskell rather than Cabal. The Heqet library still needs some work, such as implementing dynamics and polishing the time signature handling.

As predicted, the bulk of my work was on the application. I spun off the library for the dependency graph and undo into the `history-graph` library and built the framework for almost the whole application. I wish to emphasize that although the application is not usable, the code is in a much better state than it may seem. I have solved most of the hard problems of the project, but I ran out of time to implement the easy stuff. For example, many of the editing functions corresponding to currently-nonfunctional buttons are trivial to implement. A more complicated example: there is no ability to open or save files, but an important design point, the function "registry", provides the ability to encode history actions as text, since functions themselves can't be easily stored in a file. The functional reactive programming UI framework should make it straightforward to add the click-to-select functionality.

3   What I Learned

As in last semester, I've grown in my ability to find satisfaction in a large project that is not complete. I have some time off this summer before starting to work full-time, and I am very excited to continue with Heqet and make it available to users.

My skills in Haskell programming and software architecture have improved, particularly when I studied functional reactive programming.

4   Haskell

The Haskell language continues to live up to its reputation: I had almost no semantic bugs once clearing the type errors, and refactoring, such as when I added the function registry, went smoothly. My attempts to change the structure of the Heqet library's music representation didn't go well, but that was due to problems with my algorithms which would have necessitated a large refactoring that I decided to forgo in favor of focusing on the application. I didn't know what to expect from functional reactive programming, but it turned out to be simple and concise.

5   Future Goals

I now have even more open issues on Heqet's GitHub tracker: 149. In addition to the features I haven't yet implemented, there are more improvements I would like to make, such as prettier drawn music on the real-time canvas and support for multiple music notation systems, such as a text-based display suitable for screenreaders, braille music in a format for a tactile display, early music notation, and a notation system of my own design. I plan to make complete builds available for a variety of systems. I look forward to continuing with this project!

6   Images

The empty screen:
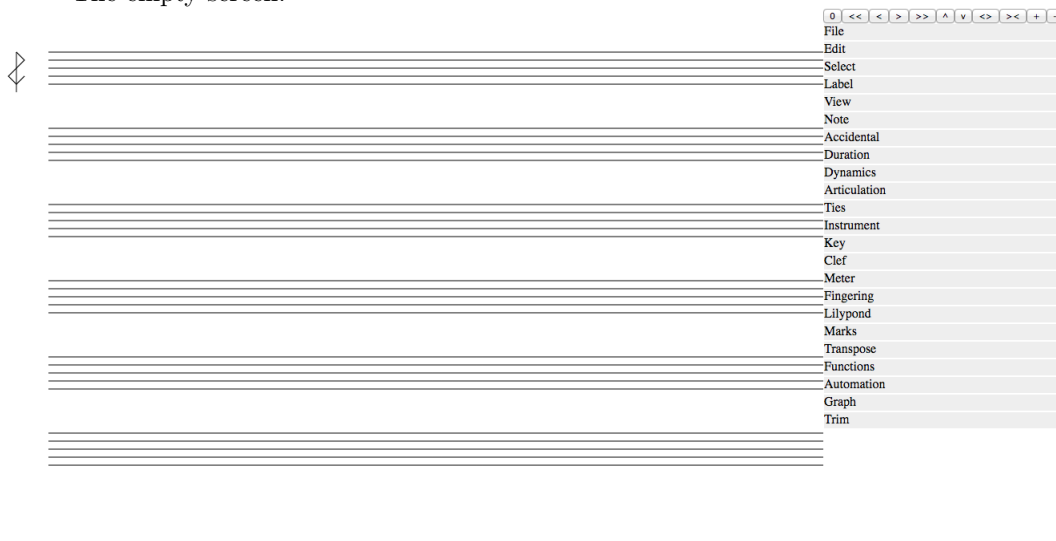
The menus on the right side, expanded:

Top row buttons: 0 | << | < | > | >> | ^ | v | <> | >< | + | -

File

[ ] Open | Save | Embed

[ ] Export Lilypond code

concert score ☐
transposed score ☐
parts ☐

Edit

copy | paste | paste without history

paste overlapping | paste only: articulation

dynamics | meter | key | clef | color

string labels | accidentals | instrument | staff

substaff | fingering | markup | lilypond | delete

copy JSON | paste JSON

create ostinato until [ ]

Select

all | none | invert

insertion point: [ ]

refine selection by:

start time | < | <= | == | >= | >

end time | < | <= | == | >= | >

duration | < | <= | == | >= | >

pitch | < | <= | == | >= | >

dynamic | < | <= | == | >= | >

time from measure start | < | <= | == | >= | >

staff == [ ]

instrument == [ ]

has lilypond: [ ]

measures at time: [ ]

highest voice on staff | middle voices on staff

lowest voice on staff | highest voice on substaff

middle voices on substaff | lowest voice on substaff

is note | is rest | is percussion | is effect | is lyric

expand selection by:

consider notes individually ☐

hold time, relax staff | hold time, relax substaff

hold time and staff, relax voice | same rhythm

hold time and substaff, relax voice | same rhythm

same rhythm on staff | same rhythm on substaff

same instrument

Label

select | deselect | color | uncolor
select | deselect | color | uncolor
select | deselect | color | uncolor
select | deselect | color | uncolor
select | deselect | color | uncolor
select | deselect | color | uncolor
select | deselect | color | uncolor
select | deselect | color | uncolor
select | deselect | color | uncolor
select | deselect | color | uncolor

remove all colors | string labels: select | deselect

add | remove

remove all string labels

View

concert | transposed

Note

A | B | C | D | E | F | G | rest | special effect

percussion: hh | sn | ...

Accidental

remove | auto-assign | bb | b | – | # | ##

Duration

sets note just entered, or all in selection

sets note just entered, or all in selection

\breve | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128

add dot | remove dot | any rational duration:

multiply by: 1/2 | 2/3 (triplet) | 4/5 (pentuplet)

4/7 (septuplet) | 8/9 | 8/11 | multiply by rational:

Dynamics

remove | fff | ff | f | mf | mp | p | pp | ppp

create cresc. | create decresc. | other dynamic

marks: fp | sfz

Articulation

remove all

add slur | remove slur

add staccato | remove staccato

add tenuto | remove tenuto

add marcato | remove marcato

add accent | remove accent

add portato | remove portato

add staccatissimo | remove staccatissimo

add stopped | remove stopped

Ties

add tie | remove tie | merge tied notes

Instrument

assign standard instrument [ ]

remove | info on instrument

Key

auto | remove | major | minor | C | Db | D | Eb

E | F | Gb | G | Ab | A | Bb | B

Clef

auto | remove | treble | alto | treble_8 | tenor

bass | manual clefs are only apparent in transposed view!

Meter

Meter

remove 2/2 2/4 3/4 4/4 5/4 6/4
7/4 3/8 5/8 6/8 7/8 9/8 12/8 x/y

x

y

sum of fractions meter

format: (a+b+c)/d + (e+f)/g

Transpose

oct up oct down by halfsteps: +1 H +2 W
+3 m3 +4 M3 +5 4 +6 tritone +7 5
+8 m6 +9 M6 +10 m7 +11 M7 −1 H 2 W
−3 m3 −4 M3 −5 4 −6 tritone −7 5 −8 m6
−9 M6 −10 m7 −11 M7 any:
by scale degrees: +1 +2
+3 +4 +5 +6 −1 −2 −3 −4 −5
−6 any:

Fingering

remove 0 1 2 3 4 5 6 7 8 9

thumb

Functions

product

Lilypond

Add function

"\fermata"

Add command

"\foo { ... }

Add post-fix lilypond item

"\fermata

Remove all matching input:

Remove all but those matching input

matching by

portamento trill ....

Automation

new notes given pitch according to their key ☐
new notes inherit key ☐
new notes spelled according to key ☐
new notes in nearest octave ☐
new notes inherit dynamic ☐
new notes inherit instrument ☐
new notes inherit clef ☐
recalculate clefs when adding a note ☐
recalculate keys when adding a note ☐

Marks

add rehearsal mark   add markup rehearsal mark

Graph

undo parents: ... redo children: ...
identity action copy history from here forward
attributes for this node...

Transpose

oct up oct down by halfsteps: +1 H +2 W
+3 m3 +4 M3 +5 4 +6 tritone +7 5
+8 m6 +9 M6 +10 m7 +11 M7 −1 H 2 W
−3 m3 −4 M3 −5 4 −6 tritone −7 5 −8 m6
−9 M6 −10 m7 −11 M7 any:
by scale degrees: +1 +2

Trim

remove history not leading to this node
remove history not leading to any of:

remove saved values for nodes, excluding:

A demonstration of various glyphs: