Heqet Tutorial

By Isaac Reilly
advised by Donya Quick
senior project for the Yale University major in Computer Science

fall 2015

Contents

1 Heqet for Euterpea Users

Given a Euterpea music expression, creating a score is straightforward. You will need to import `Heqet` and `Heqet.Input.Euterpea` and then apply the following functions: First, to convert it to a Heqet music value, you should use `fromEu` or `fromEu1`, which take a Euterpea `Music Pitch` or a `Music Note1` respectively. Now you can print a score to stdout with the functions `quickScore` or `quickLine`. `quickScore` produces a piano rendition, while `quickLine` writes a single staff.

The reason that these functions perform IO is that the simplest way to use Heqet is to write a Haskell program which outputs your Lilypond code and pipe it directly into Lilypond, which you can do by running the included `heqet.sh` bash script with your Haskell program filename as an argument. Alternatively, just load your program into GHCi and paste the output code into a Lilypond file.

Now, let's try something more complicated. Suppose you have a score that has multiple instruments and you want to write them on separate staves with appropriate labels. If your Euterpea music value has instrument modifiers, you can just use `writeScore` for your output function, since Euterpea instruments are converted automatically. If you don't have instrument modifiers, or if your instruments are beyond the basic set of instruments currently recognized, you'll have to assign instruments in Heqet. The tools to manipulate music are mostly lenses, best used with the `Control.Lens` package. Lenses and other "optics" are a way to modify portions of a larger piece of data, for example, changing the instrument assigned to some of the notes of a music value.

By the way, this is a good time to mention that in Heqet most properties of music are recorded on every single note they impact. So, every note has a field for its `Maybe Instrument`. This way, you can rearrange the music as much as you want without worrying about handling the transitions between states over the course of the music—if the instrument changes, then Heqet is supposed to automatically write this direction into the part.(Actually, this isn't implemented yet. An example that does work is handling slurs: a slurred is considered to be an articulation on all but the last note in the slur. If you cut a slurred phrase in half, both halves will stay slurred correctly.)

Suppose we have a Heqet music value `opus`. Let's make it played by an oboe:

```
opusWithInstruments = opus & traverse.val.inst .~ Just oboe
```

Quick explanation of these operators: `traverse.val.inst .~ Just oboe` is a function that can manipulate some music. `&` is a `flip`ed `$`. So we're applying this function to `opus`. In the middle is the "optic", or in this case a composition of three: `traverse`, `val`, and `inst`. `traverse` looks at every note in the music, `inst` looks at the instrument of a note, and `val` is an implementation detail that will be hidden in future versions of Heqet. The `.~` operator means to set the value revealed by the optic, in this case setting the instrument field of every note to be `Just oboe`.

2 Heqet for Lilypond Users

2.1 Note input

2.2 Differences from Lilypond input

There are several differences between the Heqet note-input domain-specific language and Lilypond input, for a variety of reasons.

- You can enter notes with any rational duration with the `\d` syntax, for example `c\d 4/5` to make a note with a duration of 4/5 of a WHAT. You can omit the denominator if it's 1. This is currently the only way to enter notes of the durations needed for a tuplet.
- `hz`
- `c-(` for slurs (sorry)
- no clefs or time signatures
- functions, commands, `\with` . Lilypond parsing problems
- only absolute entry at the moment.
- percussion notation `\phh` for hi-hat. When you enter notes, you don't need to specify that they must be rendered in a DrumStaff. They should be automatically rendered well, although support for percussion is currently minimal.
- no way to manually write beams, as this is not something the musician should need to worry about.

2.3 Making a score

2.4 Lilypond tweaks

3 Advanced topics