For string orchestra

Hans Höglund 2012

January 8, 2012

Introduction

This file contains the code for a (yet to be named) piece for string orchestra. It is a literate source file, meaning it is written both for reading and running as a program. Throughout, code will appear in a `typewriter font`. The code form a valid Haskell program that generates the piece.

If you obtain this document in source form, you can convert it to a PDF document using Pandoc[1] or compile it to a Haskell program using a Haskell compiler[2].

```
module Music.Projects.MusicaVitae
where

import Temporal.Media
```

Instrumentation and tuning

First we define the instrumentation:

- Violin I-IV
- Viola I-II
- Cello I-II

[1] Which can be found at http://johnmacfarlane.net/pandoc

[2] Such as GHC, see http://www.haskell.org/ghc

- Double Bass

```
data Part
    = Violin Int
    | Viola Int
    | Cello Int
    | DoubleBass
    deriving (Eq, Show)
```

A basic idea of the piece is to combine (slightly) different tunings of the instruments using open-string techniques and harmonics. For this purpose, the orchestra is split into three sections, each using a different tuning:

- Odd-numbered Vl, Vla and Vc parts tunes A4 to 443 Hz (A3 to 221.5 Hz)
- Even-numbered Vl, Vla and Vc parts tunes A4 to 437 Hz (A3 to 218.5 Hz)
- Double bass tunes A1 to 55 Hz

The other strings should be tuned in relation to the A-string as usual.

```
data Section
    = High
    | Low
    | Middle
    deriving (Eq, Show)

partSection (Violin 1)   = High
partSection (Violin 2)   = Low
partSection (Violin 3)   = High
partSection (Violin 4)   = Low
partSection (Viola 1)    = High
partSection (Viola 2)    = Low
partSection (Cello 1)    = High
partSection (Cello 2)    = Low
partSection DoubleBass   = Middle

sectionTuning Low    = 437
sectionTuning Middle = 440
sectionTuning High   = 443
```

```
partTuning = sectionTuning . partSection
```

All parts may be doubled. If several parts are doubled but not all, the musicians should strive for a balance between the two main tuning sections (i.e. avoid doubling just the upper parts or vice versa).

Certain cues are required to be played by a single musician even if the parts are doubled, which will be marked *solo*. These passages should be distributed evenly among the musicians, instead of being played by designated soloists.

Musical preliminaries

We are going to represent time using the *temporal-media* package[3].

```
type Pitch = Int
data Str   = I | II | III | IV
    deriving (Eq, Show)
```

Playing techniques

The piece makes use of different playing techniques in both hands. As the intonation will be different between open and stopped strings, we also define a function mapping each left-hand technique to a stopping.

```
data Stopping
    = Open
    | QuarterStopped
    | Stopped

data LeftHand p
    = OpenString p Str
    | NaturalHarmonic p Str
    | NaturalHarmonicTrem p p Str
```

[3]See http://hackage.haskell.org/package/temporal-media

```
    | NaturalHarmonicGliss p p Str

    | QuarterStoppedString Str

    | StoppedString p Str
    | StoppedStringTrem p p Str
    | StoppedStringGliss p p Str
    deriving (Eq, Show)

techniqueStopping ( OpenString           _ _   ) = Open
techniqueStopping ( NaturalHarmonic      _ _   ) = Open
techniqueStopping ( NaturalHarmonicTrem  _ _ _ ) = Open
techniqueStopping ( NaturalHarmonicGliss _ _ _ ) = Open
techniqueStopping ( QuarterStoppedString _     ) = QuarterStopped
techniqueStopping ( StoppedString        _ _   ) = Stopped
techniqueStopping ( StoppedStringTrem    _ _ _ ) = Stopped
techniqueStopping ( StoppedStringGliss   _ _ _ ) = Stopped

data RightHand a
    = Pizz a
    | Note a
    | Phrase [a]
    | Jete [a]
    deriving (Eq, Show)
```

Intonation

Many playing techiniques in the score calls for open strings. In this case intonation is determined solely by the tuning.

In some cases, open-string techniques are used with an above first-position stop. This should make the open string pitch rise about a quarter-tone step (or at least less than a half-tone step).

Where stopped strings are used, intonation is determined by context:

- In solo passages, intonation is individual. No attempt should be made to synchronize intontation (on long notes et al) for overlapping solo cues.
- In unison passages, intonation should be synchronized.

```
data IntonationType
    = Tuning
    | Raised
    | Solo
    | Choral
```

Other preliminaries

```
data Cue = Cue Part (RightHand (LeftHand Pitch))
```

```
data Phrasing = Phrasing { attackVel  :: Double
                         , sustainVel :: [Double]
                         , releaseVel :: Double
                         , staccatto  :: Double }
```