When playing at rhythmical level five, a bowed voice may switch to playing notes staccato — notes longer than 1/16 will be converted to a 1/16 note plus a rest.

All notes played by the bowed instruments are subject to a slight vibrato controlled by an ADSR envelope, that is, the finger positions are modulated by a slow sine wave, whose amplitude is governed by the ADSR.

3.3 Munola

During development, there was a need for a way of inputting music that the virtual instruments could play in order to experiment with different musical material. The composition I wanted to do required symbolic manipulation of the music, which the library implementing the notation would have to support.

One option was MIDI, but MIDI uses a binary file format that is not editable without the use of dedicated software. Also, the way MIDI represents music is not suited for the style of symbolic manipulation I wanted to do.

A textual format would be more suitable, as it would be editable using any text editor. One such system is LilyPond[3], and open source music notation language and software package. However, LilyPond is a heavy-weight package intended for full score notation and rendering and does not seem designed for integration into other projects nor symbolic manipulation. A similar project is ABC notation[4], which would be a great choice except for a few kinks in the language and the lack of a mature C++ implementation (that I know of).

To overcome the problems with existing solutions, and to enable design specific to the needs of this project, a custom notation language *Munola* (Music notation language) was created. The goal was for the language to be easy to read and write for both humans and computers, and to provide a C++ library for easy symbolic manipulation. Munola served as an excellent tool during both the technical and creative development of this project. In the end, the music is not written out in entirely in Munla, but the Munola backend is still used for symbolic representation.

White-key pitches of the equal-tempered scale are notated with upper case letters

```
C D E F G A B
```

The default octave is 4. A pitch can be raised or lowered one semitone by prepending `#` or `b`, multiple accidentals can be applied to the same note, e.g.

```
#C bC ##C bbC #bC
```

which evaluates to $C^\sharp$, $C^\flat(B)$, $C^{\flat\flat}(B^\flat)$, and just C. Accidentals are not persistent as in traditional western notation. The octave of a pitch can be raised and lowered by `^` and `_` e.g.

[3] https://lilypond.org/

[4] https://abcnotation.com/

```
^C _C _#C
```

which evaluates to $C5$, $C3$, and $C^\sharp 3$. The general octave can be selected by writing the octave number

```
3 C D E  5 F G A
```

which evaluates to $C3$, $D3$, $E3$, $F5$, $G5$, and $A5$.

The default duration of a note is one beat. The duration can be doubled using `+`, halved using `-`, and multiplied by 3/2 (dotted) using `.`, e.g.

```
+C ++C -C --C .+C
```

which evaluates to a two (2/4) beat, four (4/4) beat, 1/2 (1/8) beat, 1/4 (1/16) beat, and a three beat note.

Rests are notated using `R`. All modifiers apply to rests, but pitch modifiers have no effect.

A note can be marked as the end of a phrase using `~`, e.g.

```
C D ~E  R F G ~A
```

which are two phrases, the first ending on E and the second on A. The first note of the second phrase is a rest.

Notes can be accented by prepending `!`.

Munola supports functions which are notated by

```
function(arg1, arg2, arg3)
```

the function name cane be any string of lower-case characters. A function may have any number of arguments, including zero. Nested function calls are not supported. An example of a function employed during this project is `either`, e.g.

```
C E either(G, A, ^C)
```

which will evaluate to $C4, E4$, and $G4$, $A4$, or $C5$.