Overscore – Description of the notation

Quentin Stievenart

March 11, 2013

1 Notation

Classical notation express multiple things:

1. Each note has a duration and a pitch
2. There can be rests, which are basically empty notes with a certain duration
3. Multiple notes can be played at the same time, in a chord
4. Notes can be played while other notes are still playing

A song consists of a progression of bars (also called measures). Being able to specify bars independently might be interesting, because bars are often repeated. This would then avoid having to copy-paste bar descriptions. The song would then be represented by a list of bars to be played in the order they appear in this list.

So, songs consist of a list of progressions, each one played with a specific instruments. This allows the system to play complex songs with multiple instruments, as it is often the case in classical music.

So, the notation needs:

- Notes, played for a certain duration with a certain pitch
- Bars, composed of notes played at a certain beat
- Progressions, composed of bars played one after the other
- Songs, composed of multiple progressions each played with a specific instrument

The unit of time is the beat.

A bar can have multiple properties:

- A time signature
- A tempo

During the same song, a bar might change its tempo or time signature from the previous bars, but it is not common in classical music (no examples could be found). The notation is easily extensible to allow it if needed.

1.1 Notes

A note will be noted as follow (for a quarter-note 440-Hz A (4th octave))[1]:

```
(play :A4 1)
```

The note pitch is noted as the symbol made of the concatenation of the note itself, in conventional english notation, and the octave.

1.2 Bars

To specify when a note is to be played in a bar, the `beat` function will be called. To simplify common patterns, the functions `play-seq` and `play-chord` permits to play respectively a sequence of notes (each one playing after the previous has ended) and multiple notes in parallel.

In most songs, notes are played in sequence, and those songs do not need complex time handling features. Thus, most of the time a bar will consists of notes enclosed in a `play-seq`. The macro `simple-seq` simplifies the notation for long sequences of note, when no chord is present in the sequence.

As an example, all the following bars are equivalent:

```
(defbar foo-basic
  (beat 1 (play :C4 1))
  (beat 2 (play :A4 1))
  (beat 3 (play :G4 1))
  (beat 4 (play :C5 1)))

(defbar foo-seq
  (play-seq
    (play :C4 1)
    (play :A4 1)
```

[1] The note pitch is noted as the symbol made of the concatenation of the note itself, in conventional english notation, and the octave.

```
    (play :G4 1)
    (play :C5 1)))

(defbar foo-simple-seq
  (simple-seq 1
    :C4 :A4 :G4 :C5))
```

1.3 Progressions

A progression is a sequence of bars, where repetitions can be specified:

```
(defprog myprog
  (repeat-element 2
    foo-basic))
```

1.4 Song

A song is composed of multiple progressions played at the same time with different instruments. It also has a time signature and a tempo

```
(defsong mysong
  {:time-signature [4 4]
   :tempo 80}
  [myprog sampled-piano]
  [myprog pad])
```

1.5 Playing something

The `start` function is used to play a song:

```
(start mysong)
```

To play a note, a bar or a progression, a instrument should be specified, and the function `start-element` is used. This is useful when debugging a song:

```
(start-element foo-basic sampled-piano)
```

`start-element` also takes an optional tempo and time-signature as arguments.

2 Internal representation

Progressions, bars and notes are simply represented by a function that, when called, plays what it represents. Each of these functions take the following arguments:

- The current state (that contains the current tempo and the time signature)
- The time at which it should be played
- The instrument it should use

Each function returns the time during which it will play (in ms).

This allows easy manipulation of such elements in Clojure. Also, since all those elements share the same representation, functions like `repeat-element` are valid for notes, bars and progressions, and functions like `play-chord` could be used to play multiple bars at the same time in a progression.

Songs are represented as functions that does not take any arguments, and play its content when called.