Chordpiler User Manual

Created by Jack Anderson & Nate Romack

Overview

Chordpiler translates guitar tablature into haskell code for Tidal Cycles. The tablature can feature as many guitar chords as specified, and can optionally feature information such as beats per minute and strums for each chord. The input requires specific formatting, very similar to how guitar tablature is usually written. Chordpiler does not run the Tidal files and Tidal Cycles must be installed and ran separately to use these files. This user manual assumes the user is familiar with Tidal Cycles and knows how to run Tidal files. Chordpiler uses ANTLR which must be installed prior to use.

Input

Guitar tablature accepted by Chordpiler follows the following structure:

- Optional BPM line that specifies the beats per minute
- Any number of chords
    - Chords are optionally preceded by number of strums
    - Chords list each guitar string, along with their frets
        - Along the string, there is either a finger position listed on a fret, a mute symbol, or nothing, meaning the string is to be played open

For example:

```
test1.micro
BPM: 120

Strums: 1
e ---|---|---|---|---|
B -x-|---|---|---|---|
G ---|---|---|---|---|
D ---|-x-|---|---|---|
A ---|---|-x-|---|---|
E ---|---|-o-|---|---|

Strums: 1
```

```
e -x-|---|---|---|---|
B -x-|---|---|---|---|
G ---|-x-|---|---|---|
D ---|---|-x-|---|---|
A ---|---|-x-|---|---|
E -x-|---|---|---|---|

Strums: 1
e ---|---|-x-|---|---|
B ---|---|---|---|---|
G ---|---|---|---|---|
D ---|---|---|---|---|
A ---|-x-|---|---|---|
E ---|---|-x-|---|---|

Strums: 1
e ---|---|---|---|---|
B -x-|---|---|---|---|
G ---|---|---|---|---|
D ---|-x-|---|---|---|
A ---|---|-x-|---|---|
E ---|---|-o-|---|---|
```

Input file names must end in '.micro'.

Running Chordpiler

The easiest way to run Chordpiler is via this Micro script in a Unix environment.

```
Micro.sh
#!/bin/bash
java org.antlr.v4.Tool guitartab.g4
javac *.java
java Driver < $1
```

Place input files into the folder labeled 'inputs'. Running `./Micro.sh inputs/[input file name]` will run Chordpiler and return the output. This output can be copied over to a file ending in .tidal and opened Pulsar (or any other text editor that supports Tidal

Cycles) and ran. Channels d1-d6 each act as a string of the guitar, and must all be running to hear the chords input.

Output printed by Chordpiler follows the following structure:

- A hush statement
- A statement to set cycles per second
- Six channels each acting as an individual guitar string containing:
    - A slow statement followed by the number of chords in the cycle (to keep cycles per second in sync with beats per minute)
    - A note statement followed by the notes to be played
        - Each chord is surrounded by brackets. Inside these brackets contain the note to be played, repeated for the number of strums specified
    - A sound statement specifying the superpiano synthesizer to be used to play the notes

For example:

```
test1.tidal
hush

setcps(120/60/4)

d1 $ slow 4 $ note "[4 ] [5 ] [7 ] [4 ] " # sound "superpiano"

d2 $ slow 4 $ note "[0 ] [0 ] [-1 ] [0 ] " # sound "superpiano"

d3 $ slow 4 $ note "[-5 ] [-3 ] [-5 ] [-5 ] " # sound "superpiano"

d4 $ slow 4 $ note "[-8 ] [-7 ] [-10 ] [-8 ] " # sound "superpiano"

d5 $ slow 4 $ note "[-12 ] [-12 ] [-13 ] [-12 ] " # sound
"superpiano"

d6 $ slow 4 $ note "[~ ] [-19 ] [-17 ] [~ ] " # sound "superpiano"
```