CurvJ 1.0 Release Notes

Legal Notice

Introduction

CurvJ is a computer program developed by Harold Fortuin in Java to generate MIDI sequencer files which explore forms of ensemble coordination made possible by the computer-controlled synthesizer. Its algorithms and core functionality were developed by Fortuin in his dissertation program Curvaceous, a C language Mac/UNIX command line program released in 1993, and demonstrated that year at the Fourth International Symposium on Electronic Art.

Like Curvaceous, CurvJ can generate any number of tracks, each track being a precisely defined combination of
1) an accelerating or decelerating pulse
2) a pitch motif stated and then transposed a number of times in succession
3) a crescendo or decrescendo done with MIDI velocity values

The curves which define the acceleration/deceleration and crescendo/decrescendo are generated from hyperbolas defined by the user.

However, with CurvJ tracks can be defined from a Java Swing graphical user interface and/or from Java .properties files for ease of use. Plus, as a Java executable it can be run on nearly any operating system which supports Java, including Windows, Macintosh, and Linux.

Unlike Curvaceous, CurvJ version 1.0 does not yet contain the ability to
create groups of tracks which gradually diverge from and converge to one of these "originally defined" tracks in terms of rhythm, pitch (using MIDI pitch bend values), or both. These will be available in a future release, and will be configured from the currently disabled controls for Rhythm and Pitch Branching in the Global Parameters dialog box.

Install and Run CurvJ

1) Be sure you have the Java Runtime or Java Development kit, version 1.16 or greater, installed on your computer. Recent versions are commonly called J2SE.

If you are not sure, try typing

```
java -version
```

in your Command Prompt (or MS-DOS Command) window on your Accessories on a Windows PC, or in your Terminal Utility on Mac OS X, or from your Linux command line.

If it complains that this command isn't recognized, it is NOT installed. It can be downloaded from http://java.sun.com, or possibly from Apple for the Macintosh.

If it is installed, it should display the Java version.

NOTE: CurvJ uses the Java Swing (Java Foundation Classes) library, which was a separate download in earlier versions of Java. For such versions you need to download a compatible swingall.jar, and put this jar on your java classpath.

2) Download the CurvJAll.zip.
3) Extract into your chosen directory/folder.
4) From your Command Prompt, Terminal, or command line, navigate to your chosen directory
5) Run the command

```
java -jar {JarFileName}.jar
```

NOTE: If you downloaded a separate swingall.jar, run the following command instead:

```
java -cp {mySwingPath}\swingall.jar -jar CurvJ.jar
```

Running CurvJ

The first step required (unless you immediately quit) is to define a file for your output. From the File menu, select Save Output As, and browse to the desired directory, name the file (which, despite the file type label, CAN be read by any operating system), and click OK.

Next, select the Global Params command from the Edit menu. Here you should set your total number of tracks and tempo, and then click OK. (Later you will be able to define pitch and rhythmic convergence/divergence, also called "branching" by Fortuin.) Note that the values appearing by default can be defined in the

```
CurvJGlobal.properties
```

which should be in the same directory as the CurvJ.jar. If no such file exists, the program will present an

internally defined set of default values. (If you wish to define your own defaults, it would be wise to first copy the original file, which should then be renamed, so you will always have the original defaults.)

Next, select the Track Params command from the Edit menu. At the top of the Track Params dialog you can set the MIDI channel and see which of your ***n*** tracks you are editing (tracks can only be edited in ***1...n*** sequential order). The three tab panes, one each for Rhythm, Pitch, and Velocity, are accessible below that. Note that the values appearing by default could be defined in ***1..n***
`CurvJTrack`***`n`***`.properties`
which again should reside in the same directory as the CurvJ.jar.

Edit the values on the Rhythm, Pitch and Velocity tabs with the help of the instructions which follow. Click OK in a given Track Params dialog once you have entered all the values you want, or if you are accepting the defaults for that track. Then, if you are creating more than one track, select the Track Params command again from the Edit menu. You will need to select this command for as many times as you have tracks.

Note that once you click OK in your last Track Parameters dialog, you may see
`FINISHED Writing the MIDI File {myPath/myFile.mid}`
displayed in the command line window. (It displays on Windows if run from the Command Prompt window, but not from the Run menu. It does not display in the Terminal on Mac OS X.)

Rhythm Tab

All time values are defined in Measures, Beats, and Ticks. The program writes values to 4/4 time, and defines 480 ticks per quarter note. (Of course, 4/4 time is a useless concept from a CurvJ listener's perspective!) As shown by the tool text, time values should be entered as
measures:beats:ticks

Start Offset - how long a silence should precede the start of the first note
Target Duration - approximately how long in total the track will last
Minimum Attack - the smallest distance between subsequent note attacks that will be permitted. This value can override the Target Duration.

Other fields should be self-explanatory.

Pitch Tab

Pitches should be entered as follows, with octaves ranging from 0-10:
A-G{# or space}{space, if a #}octave
For example, C 5 G# 0 G 10

C 5 is middle C (MIDI note number 60). C 0 is the lowest MIDI note, and
G 10 is the highest.

Pitch intervals, entered for both the Interval and Transposition lists, are defined by positive or negative half-step values. For example, an octave up is 12, while a half-step below is -1.

A Segment is the pitch motif which will be repeated successively in the track in transposition. The intervals in the interval list will define the Segment. A Segment consisting of the three pitches C 5 D 5 B 4 could be defined by
1) Entering C 5 as the Starting Pitch
2) Entering 2 in the Interval field, and then clicking the + Interval button
3) Entering -3 in the Interval field, and then clicking the + Interval button

If a mistake was made in these entries, the - Intervals button can be clicked to clear the interval list. The - Transp's button works similarly for transpositions.

The transposition interval values in the transposition list will define the distance between the final note of last Segment and the first note of the start of the subsequent transposed Segment. So to repeat our example Segment up one half-step after its first statement, a 2 (since it would need to start 2 half-steps above the last pitch of the previous Segment) should be entered in the Transposition field, and + Transp should then be clicked. Of course, a number of additional transpositions could be entered this way.

The program will loop through its list of transpositions until all notes have a pitch. And of course the final Segment can be truncated if the number of notes resulting from the Rhythm values is not an even multiple of the number of pitches in the Segment.

If the pitch calculations result in a pitch that is higher than the Upper Bound, or lower than the Lower Bound, the interval between the Bound and the out-of-bounds pitch is calculated

If Reflect From Bounds is chosen:

1) The calculated interval is multipled by -1 and added/subtracted to the crossed Bound to result in an in-bounds pitch
2) All intervals on the inverval and transposition lists are multiplied by -1 and added in order to their respective previous pitches, until another Bound is crossed.

If Wrap Around Bounds is chosen instead of Reflect from Bounds:
1) The calculated interval is multipled by -1 and added/subtracted to the opposite Bound to result in an in-bounds pitch near the opposite Bound.
2) All intervals on the inverval and transposition lists are then applied as usual, until another Bound is

crossed.

Velocity Tab

The Largest and 2nd Largest Velocities define the MIDI velocity curve. Values must range from 1 - 127.

Samples

Some of the original Curvaceous examples have been ported to CurvJ, and can be executed from their respective directories.

StringHyp: a series of acceleration-decelerations on one MIDI channel, done with successively higher time offsets on copies of the same track.
StringHypCanon: like the previous, but with a group of tracks comprising a second line of similar music.
StringHypCanonInverted: like the previous, except now that the second line of music has inverted intervals.

Future Releases

The ability to create groups of tracks which gradually diverge from and converge to one of the "originally defined" tracks in terms of rhythm, pitch (using MIDI pitch bend values), or both will be implemented and available from the Global Params dialog.