

# Lindenmayer Tool

---

## **Audio Extension for Lindenmayer-System Model Exporter**

*A better name probably exists.*

Michael Gircys

COSC 4P98 - Topics in Computer Media and Digital Audio

Prof. Brian Ross

2014-05-09

## Notice

This document outlines new audio features for the existing Lindenmayer-Tool model exporter project.

The original documentation and source code is available in the [Documentation/Previous Documentation](#) subfolder.

## Requirements

This program was created using Microsoft's .NET framework version 4.0 on Windows 7 x64. While care was taken better allow successful operation on varied systems, the program has not been tested outside of the noted build environment.

Live audio playback of the interpreted L-System relies by default on the Csound binary directory being included in the system PATH variable. If this is not the case, the Csound executable filepath can be specified through the program's settings window.

## New Features

A number of key changes have been made to the tool to allow for the interpretation of L-Systems as an audio event source, and to export these events in a Csound score format.

The top menu bar has been adjusted to allow for a separation of graphical and audio commands. The vast majority of the new features will be found within the Audio submenu.

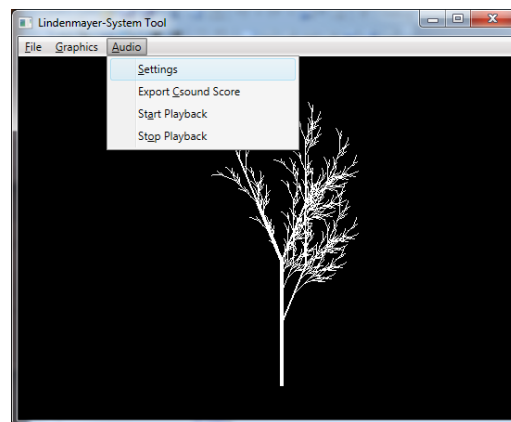


Figure 1 - The main program display and updated menu

The *settings* menu option will open a window which presents all the options that are available for tweaking the audio interpretation of the L-System. While defaults should allow for export of the provided example .lsys files, configuration will likely be necessary to produce more desirable sounds and to handle more complex L-Systems. Settings will be discussed in details shortly.

The *Export Csound Score* command presents a Save File dialog, and will then attempt to export a set of computed audio note events into a Csound score format file. These audio events are computed immediately after first loading the L-System, or after manually requesting to recompute the L-System. Stochastic systems will be interpreted as they appear in the preview window.

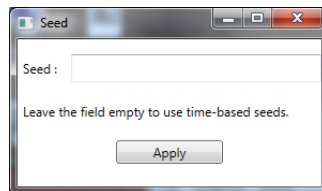


Figure 2 - RNG Seed specification dialog

A dialog is available to configure the random number generator with a specific seed for stochastic systems. Once a new seed has been specified (or removed) the L-System will be recomputed using the specified seed, or default time-based seeds if no seed value is found. As before, the Refresh command will recompute the loaded L-System.

The audio score computed from the L-System can be played as an audio stream by selecting the *Start Playback* command.

When audio playback is requested, a Csound score will be exported to a temporary directory, where it will then be compiled by Csound using a specified orchestra file to produce a wavefile. After the Csound executable has generated the wavefile, the wavefile will be opened and played. If the compilation was successfully generated, the preview window will update, highlighting the active parts of the graphical interpretation of the L-System which correspond to currently playing audio events while the audio file is playing.

Audio playback can be stopped by selecting the *Stop Playback* command.

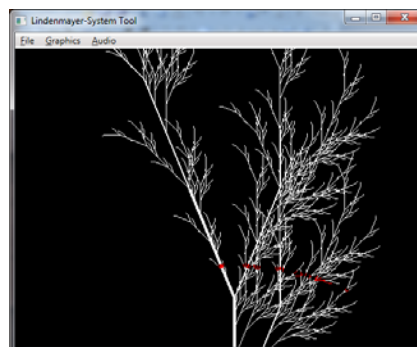


Figure 3 - Playback with active events highlighted

## L-System Interpretation

Audio events are generated through interpretation of the l-system by including an alternate mapping of symbols. A LOGO-style graphics turtle updates state based on the stream of characters it gets passed – the result of the l-system. In addition to performing graphical actions that were mapped to the incoming characters, the character stream is decoded with an alternate mapping which handles audio-related member and state changes.

Audio related state is updated either directly by processing characters which map to various update commands, or by binding graphical state to audio fields (i.e. position coordinates to pitch). A list of the audio interpretation of valid commands can be found near the end of this document.

## Score Format

The following fields are used for each event stored in the score file in the order presented:

Instrument	-	The instrument number. Preceded with “i” to follow Csound i-code expectations
StartTime	-	The time in seconds at which the event starts
Duration	-	The duration of the event
Pitch	-	* Pitch of the event (Defaults present pitch as a midi pitch index)
Volume	-	* Volume of the event
Custom1	-	* Custom field
Custom2	-	* Custom field
Custom3	-	* Custom field

\* Suggested use. Fields may be remapped by custom instruments

A number of example l-systems have been provided in the `../Examples/` and an example Csound orchestra and score header can be found in `../Examples/Sound/`.

## Settings

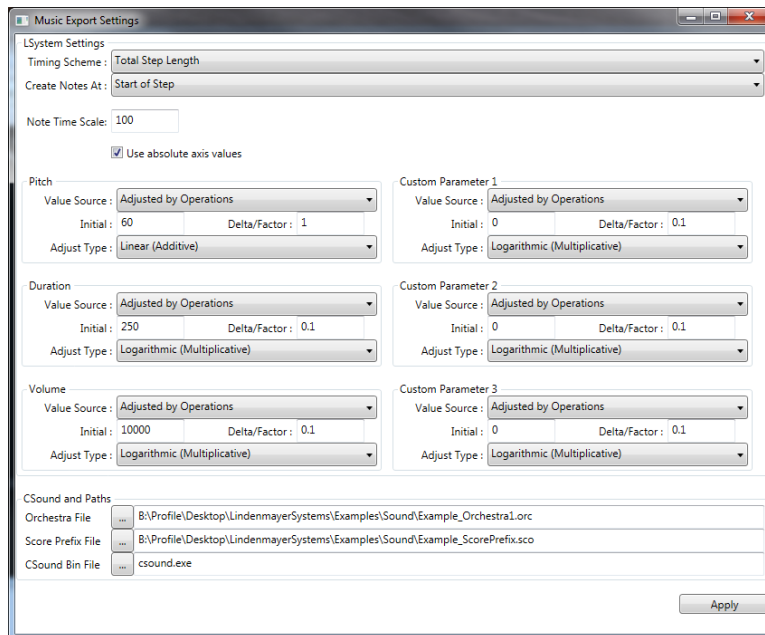


Figure 4 - Settings window with defaults

### L-System Settings

Timing Scheme - A number of different options are available for determining the starting time of interpreted sound events. The starting time is a factor of one of the following values:

- Total Step Length – Total steps taken by the turtle from root node to the current position
- 3D (XYZ) Origin Distance – Euclidean distance from the origin
- 2D (XY) Origin Distance – Distance along the XY plane from origin to the current position
- X Axis Position – X coordinate value
- Y Axis Position – Y coordinate value
- Z Axis Position – Z coordinate value

The starting time of each event is taken by multiplying the value from the timing scheme above with the number of milliseconds each step requires before advancing. The time required for each step is specified in the *Note Time Scale* field.

Playing the tones between two pitches is not typically as useful as drawing between two points. While slurred notes or a transition between pitches is possible, a single note of consistent pitch is much more desirable for an atomic audio event. Whether the audio event is created before updating the turtle or after updating the turtle is configurable from the *Create Notes At* drop-down box.

## *Individual Parameter Settings*

Each parameter beyond instrument and start-time has a number of options defining how its value is calculated.

The source of the parameter, or *Value Source*, determines which variable is applied to a formula or continuously adjusted to calculate the parameter's value. Options are:

Adjusted by Operations – Internal values updated by character commands (listed near the end of this document).

X Axis Position – Turtle position X coordinate

Y Axis Position – Turtle position Y coordinate

Z Axis Position – Turtle position Z coordinate

The *Initial* field sets a value which is the default value of operation-adjusted values, or an added constant to axis-based values (post-scaling).

The *Delta/Factor* field determines how much a value is changed per operation, or a scaling factor for axis values.

The *Adjustment Type* option determines whether, as a result of an operation, the delta is added to the current value or the delta is used as a fractional percent by which the value is incremented or decremented. (I.e. add 0.1 units to the value per operation, or add 10% per operation).

The delta is used as a scaling factor if the source is an axis coordinate and the adjustment type is not linear.

## *Csound and Paths*

The second group of options on the settings window relate to the compilation of Csound files for live playback.

*Orchestra File* specifies the Csound orchestra file to use when generating a wavefile from the interpreted score.

*Score Prefix File* specifies a partial Csound score file for header information. A custom orchestra file will likely rely on computed data tables which are declared in a score file. The score exported from the program only contains note events; it is instrument agnostic and so does not export any function tables that might normally be required. To allow for more complex instruments to be used automatically with the built-in playback options, score file headers – including function table declarations – can be stored in a separate file to be prepended to the score before Csound compilation.

*Csound Bin File* specifies the path of the executable Csound binary file. If the Csound bin directory is included in the system path, only the filename is required.

## L-System Symbol Interpretation – Graphics and Audio

<u>Symbol</u>	<u>Graphical Interpretation</u>	<u>Audio Interpretation</u>
F	Move forward and draw a line.	Advance time and create note event.
f	Move forward without drawing a line.	Advance time silently.
+	Turn left.	Increment audio pitch.
-	Turn Right.	Decrement audio pitch.
^	Pitch up.	Increment custom parameter 1.
&	Pitch down.	Decrement custom parameter 1.
\	Roll left.	Increase note duration.
/	Roll right.	Decrease note duration.
	Turn around.	Switch audio pitch decrements to increments, and vice-versa.
\$	Rotate the turtle to vertical.	Restore audio pitch behaviour.
[	Start a branch.	Start a branch (play concurrently).
]	Complete a branch.	Complete a branch.
!	Decrement the diameter of segments.	Reduce volume.
'	Increment the current colour index.	Rotate through instruments.
%	Cut off the remainder of the branch.	Cut of the remainder of the branch.
{	*Start a polygon.	Increment custom parameter 2.
}	*Complete a polygon.	Decrement custom parameter 2.
G	*Move forward and draw a line. Do not record vertex.	--
.	*Record a vertex in the current polygon.	Decrement custom parameter 3.
~	*Incorporate a predefined surface.	Increment custom parameter 3.

\* These commands have not been implemented for the graphical engine

## References:

Lindenmayer, Astrid, et al. *The Algorithmic Beauty of Plants*. New York: Springer, 1990. Print.

Wavefront Technologies. *Object File Specification*. Accessed 2013-12-18.

<<http://www.martinreddy.net/gfx/3d/OBJ.spec>>

“Nick”, *Response: How to bind an enum to a combobox control in WPF?* Accessed 2014-05-01

<<http://stackoverflow.com/questions/6145888/how-to-bind-an-enum-to-a-combobox-control-in-wpf>>

Shiskin, Sergei. *Microsoft patterns & practices - Composite Application Guidance for Windows Presentation Foundation – ValueDescription*. Accessed 2014-05-01

<<http://code.google.com/p/messagingwpf/source/browse/trunk/src/StockTraderRI/StockTraderRI.Modules.Position/Orders/ValueDescription.cs?r=4>>

Soddell, Jacques, and Soddell, Fran. *lsys2midi*. Accessed 2014-05-09.

<<http://cajid.com/jacques/lsys/mapping.htm>>

Boulangier, Richard. *Introduction to Sound Design in Csound*. Accessed 2014-05-08.

<<http://www.csounds.com/chapter1/index.html>>