

Introduction

AudioMulch is an interactive musician's environment for Pentium and Pentium II computers running Windows 95/98 and Windows NT. Bringing together the popular with what has up to now been considered experimental, AudioMulch merges the worlds of mainstream electronica and electroacoustic sound composition to create a fluid sonic environment only limited by the artist's imagination.

While many of the processes featured within AudioMulch are not new to computer music programs, it is the software's ability to carry out these traditionally "studio" or "non-real-time" signal processing techniques in real-time that emerges as its major asset. With AudioMulch, music that has for so long been limited to the pre & post-production practices of a studio can emerge in a new LIVE and interactive context.

Through the essentially unlimited (*only by the power of the computer*) combination of a network of synthesis and processing contraptions, AudioMulch allows the user to extend their current audio processing capabilities or create new music within their computer without additional expensive software or hardware systems.

Above all, AudioMulch is ***a musician's system.***

System Requirements

AudioMulch is a Win32 application requiring either Windows 95/98 or NT 3.51 (or later). To hear audio in real-time (that is what the program is designed for) you need a soundcard capable of delivering 16bit 44.1k stereo sound. To process audio from the soundcard input you will need a full-duplex soundcard i.e. one capable of simultaneous 16bit 44.1k stereo recording and playback. DirectSound is neither required nor used in the present version. To do anything useful a Pentium(R) class computer is required. MMX won't speed things up much but enhanced floating-point (Pentium Pro or Pentium II) will. The faster the machine, the more you can do in real-time.

What's New in Version 0.8b3

Version 0.8b3 fixes many bugs from version 0.8b2 and adds four new contraptions: RissetTones, RissetFilters, Nebuliser and Arpeggiator. MIDI support has been expanded to allow MIDI controllers to map onto sub-ranges of parameters, it is also possible to specify non-linear mapping curves. Soundfile support has been enhanced, with the addition of a preview function in the open soundfile dialog box. A number of other minor changes have been made.

The appendix contains a **changes** topic, listing a full history of changes for this and all previous versions of AudioMulch.

Future Developments

I am developing AudioMulch in my spare time. As a result, the rate at which new features are added to the program will vary. I have listed planned developments for AudioMulch up to and including version 1.0 below. This list is provided for informational purposes only, I make no guarantees as to its validity. The order of contraption implementation is somewhat arbitrary. If you have any comments, please email at: rossb@audiomulch.com.

Later 0.8 Versions:

- An expanded set of example files

Additionally the following contraptions will be added (others are also in the works):

- Play list based multiple wave file player
- 4-pole resonant filter with envelope follower
- Pulsar sample and hold comb filter
- FM Oscillators

Version 0.9:

- Parameter and preset recording, sequencing and automation
- Improved pattern editing in Drums and Bassline
- Definable punch in and out points for wave file in SoundIn
- Batch soundfile processing
- Undo

Additionally the following contraptions will be added:

- Simple units such as oscillator, State-variable filter, Single comb filter, low and high-pass filter
- Flexible multi-tap delay lines
- Risset tone oscillators and filters
- Enharmonic additive oscillator
- Granular filter
- Compressor / limiter / noise-gate
- Stereo chorus
- Multi-waveform oscillator
- Mixer with fx sends and mutes
- Sampler / Triggered loop player

Version 1.0:

- Support for DirectX plugins
- Support for complex automation sources such as lfos

Additionally the following contraptions will be added:

- Wave-terrain oscillator
- Pitch shifter
- Improved reverbs
- Variable speed wave file player

Registering AudioMulch

AudioMulch was created to assist in both computer music education and creative music making using computers. For this reason I have not crippled or restricted unregistered versions of AudioMulch.

However, this project has taken over a year of my spare (and not so spare) time, and I ask those of you who have the means to purchase a shareware license. A single-user AudioMulch license may be purchased for US\$50 that will register the program's future releases up to but not including version 2.0. Upon receipt of payment a registration code will be supplied which disables shareware warning messages and beta version expiry.

To register AudioMulch choose **Register AudioMulch** from the **Help** menu. A wizard will guide you through the registration process. You can also register online via the AudioMulch web site.

All transactions with regard to AudioMulch registration fees will be handled by Kagi, a shareware payment processing service, making it easy for people to pay shareware fees. You can find out more about Kagi at their web site: <http://www.kagi.com>

If you have not yet registered AudioMulch through Kagi, the Kagi Register program will take your details and provide you with a number of convenient payment options. Upon receipt of your order Kagi will forward your registration number.

Overview

Introduction

AudioMulch is an interactive music studio that allows its user to create myriad environments for the synthesis, processing, and assembly of musical components or sounds. The combination of integrated synthesizers and control mechanisms (contraptions) that make up AudioMulch facilitate the transformation of sound to create sonic textures beyond those traditionally associated with instrumental music. Importantly however, AudioMulch maintains the potential for Real-Time performance, a feature lacking in much of what has up to now been described as computer music or sonic art. Based on the notion of controllable sonic flow, AudioMulch passes externally created sources or internally synthesised textures through a range of sound creating and editing components, many of which have played a prominent role in the repertoire of 20th century music technology. By altering the paths of these input sources and interchanging contraptions the sonic composition of any piece may be transformed in a process which is perhaps best described as designing sound. AudioMulch is a musician's tool and its products should therefore not be confined within the limitations of abstract sound pieces. Easily combined with live performers in an interactive context or used to create a range of popular electronic musical styles, the depth and range of AudioMulch's features mean that creation is only limited by the power and specifications of your machine and the scope of your imagination.

Split into sections, AudioMulch's components can be generally categorised under the sub-headings ***Signal Generators, Effects, and Filters***. In addition to these central features there are a number of contraptions devoted to the channeling of sound into, through, and out of the network. Sound can be drawn through a direct line into the sound card, such as a microphone input, or can be replayed internally as WAV files for further processing. As the software works in Real-Time the sonic output can be heard directly or can be saved as WAV files to be replayed later.

Signal Generators

AudioMulch includes a range of contraptions devoted mainly to the creation of sound. From SoundIn and LoopPlayer, both devoted to the importing of sound as WAVs or through a direct line, to Bassline and Drums, contraptions commonly associated with popular electronica and earlier analog and sampling technologies. As a result of these inclusions music can be composed using entirely computer based input, whether it be from files stored on the computer's hard disk or from sounds synthesised entirely within AudioMulch.

Effects and Filters

To complement its Signal Generators, AudioMulch provides a large number of Effects and Filter contraptions that may be used to transform and shape the input signal. Some of these, like SDelay, NastyReverb and Phaser, will be recognised as common effects utilised both in the studio and on stage. Others like 5Combs and DLGranulator allow easy access to computer music techniques previously limited to the creation of recorded pieces. Offering Real-Time processing, AudioMulch opens up these technologies to a new level of use and offers non-experts the chance to play in new ways with sound and music. The range of processes available is not however limited to those found within these categories as the software also supports the industry standard VST (Virtual Studio Technology) plugin format giving AudioMulch users access to a large number of third-party processes.

Parameter control through MIDI means that AudioMulch is not confined to use with Mouse and Keyboard and can grow beyond the traditional PC setup. While not every one of the features offered by AudioMulch are new, the software's innovative composition broadens the potential use of computers within all forms of music and opens the door to a new way of thinking about sound. It's simply a case of keeping an open mind and letting your ears explore.

The next section will hopefully set you on your way.

Navigating the User Interface

The Patcher and Property Panes

AudioMulch opens with a single window split into two panes.

The pane on the left is the **Patcher pane**. You connect **Mulching Contraptions** together in this pane.

The pane on the right is the **Property pane**. Most Mulching Contraptions in the Patcher pane also have a corresponding **Property Editor** in the Property pane (Note: both *busses* and *AuxIn/Outs* do not have their own property editors, working automatically or in response to settings made elsewhere).

The Property Editor for a Mulching Contraption can be made visible by double clicking on the Mulching Contraption. To make the Property Editor invisible, click on the close box of the particular Property Editor.



Patcher pane

Property pane

Mulching Contraptions

To add Mulching Contraptions to the Patcher pane, you use the context menu. Right-click in the patcher pane to reveal the context menu.



When you select a Mulching Contraption from the context menu it subsequently appears within the patcher pane.

The Mulching Contraptions fall broadly into eight categories that are reflected in the context menu in the Patcher pane:

- **Input/Output** contains contraptions that control sound output and sound input.
- **SignalGenerators** are contraptions that generate sounds that can be further processed by other contraptions.
- **Effects** are contraptions that process the audio signals from other contraptions.
- **Filters** fulfil a similar role to effects in that they process signals from other contraptions.
- **Busses** are contraptions that allow multiple input signals to be sent to a single output or a stereo pair.
- **Mixers** are similar to busses except that they feature individual volume controls for each input pair.
- **Beta** contains new and developmental contraptions that, as a result, may not be completely bug free. The Beta section features a range of contraptions that would otherwise be placed in one of the above categories.
- **VST Plugins** contains external modules compatible with the Virtual Studio Technology features of certain other sound processing and sequencing software alternatives. VST plugins can be downloaded from a variety of locations on the Internet. A few relevant VST links are provided on the AudioMulch web site. All VST-compatible modules, once plugged-in, will appear within this category of the context menu.

Inputs and Outputs

To hear any sound, it is essential to have a SoundOut contraption in the Patcher pane. This contraption represents the audio output of Audio Mulch. All other contraptions must directly or indirectly connect to the SoundOut to be heard.

SignalGenerator contraptions have outputs that appear on the bottom of the contraption. These contraptions generate sound but don't process sounds from other sources and consequently, they don't have inputs.

Effects, Filters, Busses and Mixers have inputs on the top of the contraption and outputs on the bottom.

Mulching Contraptions are connected together by dragging a patch cord from the output of one contraption to an input of another.

Multiple lines may be taken from any output and joined to any input. When multiple cords are joined to an input the signals are automatically mixed. In more complex patches, busses or mixers can be used to combine multiple outputs into a single input.

Real-time Audio and Clock Control

In order to make any sound with AudioMulch, real-time audio must be active. Real-time Audio is controlled by the Enable Audio menu item in the Control menu, or toggled with the speaker button on the toolbar. To process real-time audio input the Use Input File option of the SoundIn contraption must be disabled.

Some contraptions are clock based (drum machine and bassline for example), and require that the clock be started as well. The clock can be started and stopped from the Control menu or the play (arrow) and stop icons on the toolbar. Tempo is also controlled from the toolbar. If you start the clock, real-time audio will be enabled automatically and the enable audio button will reflect this.

Guide to the Example Files

A number of example documents demonstrating different aspects of AudioMulch's application are provided within the Examples directory. A particularly good starting point for experimentation with AudioMulch, the examples offer simple contraption networks for you to practice on. By moving the knobs and sliders each of the individual contraption parameters can be changed - thereby allowing the user to experience the real-time sonic results of different modifications. The example documents also allow you to connect new contraptions or dismantle each document to examine its individual parts. Each example file is described below.

In order to hear any of these examples press the enable audio button on the toolbar or select enable audio from the control menu. Some examples require the clock to be running - this can be achieved by pressing the play button on the toolbar or selecting play from the control menu.

Although these examples mainly demonstrate the capabilities of AudioMulch to synthesize sounds they could be easily modified to take their input from a soundfile using SoundIn and used to process and transform other recorded sound sources.

Harmonics

The harmonics example consists of the following serial chain of contraptions: 10Harmonics->Flanger->DLGranulator->SDelay. You can read more about the individual contraptions by following the links.

This document used a 10Harmonics contraption to generate a vocal like tone as a source for sound processing. This tone is then fed through a flanger that slowly sweeps up and down, accentuating the harmonic series of 10Harmonics' output as it goes. This signal is fed into a DLGranulator (delay line granulator) which breaks it up into individual events (grains) each transposed down about an octave (using the Trans parameter) and each drawn from the source with a random delay. If the delay parameter were not set across a range, the harmonics of the event would follow the contour of the flanger (try this by setting both thumbs of the delay parameter to minimum) but instead each event has a randomized harmonic colouration. The DLGranulators output is fed into a SDelay (stereo delay line) to create a rhythmic delay effect.

SimpleSpat

The SimpleSpat document demonstrates the use of the Sspat (Stereo Spatialiser) contraption. A signal (in this case white noise filtered into a coloured band using MparaEQ) fed into Sspat may be processed to create the effect of a sound passing through space. Sspat has a separate reverb send output that may be used to increase the depth of sounds as they move away from the listener - in this case the reverb send output is fed into the NastyReverb reverb unit. The mixer (S4Mixer_1 - a four stereo input mixer) may be used to adjust the relative levels of the different signals, or to isolate different signals (try listening just to the filtered noise before it enters sspat by turning down all mixer channels except 3&4).

Technoid

Technoid demonstrates one simple way of making techno music with AudioMulch. A Bassline is fed through a stereo delay line configured to create rhythmic delays, the delay output is mixed with the output of a Drums drum machine using a stereo buss. The balance between the Drums and Bassline may be varied using each contraption's volume control.

Trance1

Trance1 is another techno example, this time utilising two basslines – one processed with a stereo delay, and another using a flanger.

Overview of AudioMulch Contraptions

The Mulching Contraptions fall broadly into six categories that are reflected in the context menu in the Patcher pane:

- **Input/Output** contains contraptions that control sound output and sound input.
- **SignalGenerators** are contraptions that generate a sound that can be further processed by other contraptions.
- **Effects** are contraptions that process the audio signals from other contraptions.
- **Filters** fulfil a similar role to effects in that they process signals from other contraptions.
- **Busses** are contraptions that allow multiple input signals to be sent to a single output or a stereo pair.
- **Mixers** are similar to busses except that they have individual volume controls for each input pair.
- **Beta** contains new and developmental contraptions that, as a result, may not be completely bug free. The Beta section features a range of contraptions that would otherwise be placed in one of the above categories.
- **VST Plugins** contains external modules compatible with the Virtual Studio Technology features of certain other sound processing and sequencing software alternatives. All VST-compatible modules, once plugged-in, will appear within this category of the context menu.

Double clicking on the contraption within the patcher pane will open the contraption's user interface or properties editor. This editor will appear as a window within the Properties pane. Adjusting the parameter controls within each window will alter the composition of the sound output. Each properties window is headed by a title bar featuring the name of the contraption and three small icons. The question mark (?) when clicked provides access to the section of this Help file directly relating to that particular contraption. The Contraption reference can be found in its entirety in the following sections. To the left of the help button is the Preset button. A complete explanation of contraption preset settings can also be found in a following section. Finally, the cross to the right of the Help button will hide the contraption's property editor. Unlike other Windows applications the cross (X) does not delete the contraption it merely hides its editor. To reveal a property editor double click on the contraption in the patcher pane. The window will reappear with its settings intact. A contraption will only be removed from the document if it is deleted in the patcher pane.

Using Common Contraption Controls

Knob

Knobs are used to control a single parameter value. To change the value of the knob click and drag the pointer up and down on the control - dragging upwards increases the value, and dragging downward decreases the value. This same method is also applicable to **increment/decrement control arrows**, such as those found on **SDelay's** Delay time editor.

Slider

Sliders are used to control a single parameter value; they appear in both horizontal and vertical orientations. To change the value of the slider click and drag the pointer on the control; up and down for vertical sliders, left and right for horizontal sliders. Vertical sliders have the minimum value at the bottom and the maximum value at the top while horizontal sliders have the minimum value at the left and the maximum at the right.

Range Slider

Range Sliders function similarly to Sliders except they are used to control the minimum and maximum of a range of values. When you click and drag on a Range Slider both the minimum and maximum values are moved simultaneously in the same direction. To move the minimum and maximum values independently you simply need to click the pointer directly over the desired slider. This can also be achieved using the keyboard. Holding down the shift key while dragging will move only the maximum indicator while holding down the alt key while dragging will move only the minimum indicator. When both shift and alt keys are held down simultaneously the minimum and maximum indicators move in opposite directions.

Important Note

When making minimal alterations to parameter or range values hold down the control key when you click and drag the pointer over either the knobs or sliders. This will slow down the movement of the individual controls allowing more exact values to be achieved.

Using Contraption Presets

Each Mulching Contraption may have up to 20 presets associated with it. A preset is a snapshot of all parameter settings for a single contraption. Presets allow you to store and recall multiple configurations - like drum patterns or filter settings.

■

Presets are accessed from the presets button positioned next to the help button of each Contraption's property editor window. Clicking on this rollover button will display a popup menu showing 20 presets. The enabled items in the presets menu contain previously stored settings. To restore a previously stored preset select it using the left mouse button. To create a new preset, right click on the preset number you wish to store to and select Store from the popup menu. Presets can be cleared by right clicking on the desired preset and selecting Clear from the popup menu.

While using a contraption the presets button indicates the current preset, whenever you edit a Contraption parameter the preset button will display '--' indicating that no preset is currently active. Presets are only stored by explicitly storing them using the presets menu.

SoundIn

Category: Input/Output

Inputs: none

Outputs: left, right

The **SoundIn** contraption is the primary source of incoming sound within many AudioMulch documents. In particular it lends itself to real-time processing of both external elements (i.e., live instrumental performances channeled through the soundcard audio input) and extended pre-recorded sound segments, in the form of wave files. While any AudioMulch document is unable to support more than a single **SoundIn** contraption, the **Use Input File** checkbox allows the user to switch between the soundcard audio input and an input wave file.

When using the soundcard audio input (**Use Input File** checkbox clear), Audio Input must be started using the **Controls->Enable Audio** menu item or the speaker button on the toolbar. When using a wave file, input is controlled using the **Play/ Stop** button on the **SoundIn** properties editor.

The remaining controls for **SoundIn** are described below;

File

When **Use Input File** is enabled, the file panel indicates the source wave file for **SoundIn**. Files may be selected by clicking on the open button next to the file panel. Importantly, at this stage, AudioMulch is only able to support 44k 16bit wave files.

File position Trackbar

The **file position trackbar** can be used to select the current playback position from the input file. It can be used while a file is playing or when a file is paused.

Auto Rewind

When checked, **Auto Rewind** causes the input file to be rewound to the beginning each time it is played.

Loop

When checked, **Loop** causes the input file to be played continuously, beginning from the start when the end has been reached.

Delay

When **Loop** is enabled, **Delay** controls the length of silence inserted between the end and the start of the input file as it is looped.

Play/Stop Button

The **Play / Stop** button starts and stops playback of the input file.

SoundOut

Category: Input/Output

Inputs: left, right

Outputs: none

The **SoundOut** contraption is the primary channel for outgoing sound in an AudioMulch document. Contraptions connected to the **SoundOut** will be heard from the soundcard output. Real-time audio is only generated when the Audio Output is active, this can be achieved using the **Controls->Enable Audio** menu item or by using the speaker button on the toolbar.

The **SoundOut** contraption also allows the output sound to be simultaneously recorded to a wave file. The controls for recording a wave file with the **SoundOut** contraption are described below.

File

The file panel indicates the wave file that **SoundOut** records to. Files may be selected by clicking on the open button next to the file panel. New files may be created, or existing files selected.

Record Mode

Record Mode determines whether any existing sound in the selected output file is overwritten or appended in subsequent record operations. Append would be useful for accumulating multiple takes for later editing while overwrite will destroy any existing sound in the file.

Punch Mode

Punch Mode is used to determine when to start and stop recording. In **manual** mode, recording begins when the record button is pressed and stops when the stop button is pressed. In **SoundIn Sync** mode, recording begins when the SoundIn is started and stops when the SoundIn stops, or when the stop button is pressed. This mode may be useful for producing a processed version of an existing wave file. **Timed** mode begins recording when the record button is pressed and continues for the time specified or until the stop button is pressed, whichever comes first.

Record / Stop Button

The **Record / Stop** button controls the commencement and termination of recording to the output wave file.

Play / Stop Button

The **Play / Stop** button allows the output wave file to be auditioned. The file is always played from the beginning.

AuxIn/AuxOut

Category: Inputs/Outputs

Inputs: left, right (*only on AuxOut*)

Outputs: left, right (*only on AuxIn*)

AuxIn and **AuxOut** are a group of input and output contraptions provided for those users running multi-channel soundcards. Using these contraptions separate channels within AudioMulch can be assigned to specific channels of the soundcard. Unlike SoundIn and SoundOut however, AuxIn and AuxOut contraptions have no property editor so all settings must be adjusted within the AudioMulch settings dialog box. This can be accessed through the *Edit* menu.

A full explanation of this procedure can be found within the *Settings Dialog* section of the **User Reference** chapter.

TestGen

Category: Signal Generators

Inputs: none

Outputs: mono

TestGen is a simple sine tone and white noise generator. The output may be switched between these two using the radio buttons. Particularly helpful when exploring the myriad textures produced by a variety of alternate contraption chains, **TestGen's** ability to generate sine waves of varying frequency provides a reasonable exploratory field in which to assess the effect of most contraptions on a range of more complex sounds. The frequency of the sine wave can be typed directly into the field provided or swept up and down with the mouse by clicking and dragging on the up/down button to the right of the field. The accepted frequency range is 20 to 20000 Hz.

The use of **TestGen** is not however restricted to the process of familiarisation. For example: the noise source may be used on its own or mixed with other sounds prior to filtering as a means of thickening or making more noticeable any filtered effects.

Drums

Category: Signal Generators

Inputs: none

Outputs: mono

Drums is a clock synchronised drum machine which uses wave files as sound sources. Five samples may be loaded at once, each with its own volume, mute and two bar sixteenth note sequencer. Channels 1 and 2 may be gated against each other for 'closed gates open' high-hat sequencing. **Drums** will not output any sound if the clock is not running.

Parameters

Volumes

The leftmost volume knob controls the overall output volume of **Drums**, the subsequent numbered knobs individually control the gain of each drum channel.

Gate 1 & 2

When checked, **Gate 1 & 2** prevents both drum channels 1 and 2 from being heard simultaneously. If channels 1 and 2 both have a note sequenced on the same pulse, channel 1 is heard. This is primarily intended for creating the effect of a closed high-hat terminating the ringing of an open high-hat. Typically the closed hat sound would be placed in drum channel 1 and the open hat sound in drum channel 2.

Mutes

The checkbox labeled **Mute**, controls muting of **Drums** as a whole, while the checkboxes at the left of each drum channel allow muting of drum channels individually. In all cases muting begins or ends at the beginning of the next bar.

Drum Channels

The lower part of the **Drums** editor is populated by five rows of identical controls. Each row pertains to drum channel, the number of which is indicated at the left of each row. Next to the channel number is a mute checkbox (described above), following that is the wave file selector / indicator. This button indicates the current wave file being used for that channel, it may be pressed to load a different wave file. The wave file selector / indicator displays "[empty]" when no file has been selected, an exclamation mark (!) next to the file name indicates that the file does not exist. Although selected wave files may be longer, **Drums** will only load up to one second of sound for each channel. Following the wave file selector / indicator is a sequence grid representing two bars of semiquavers. Beat numbers are indicated across the top of the grid. Sequence grid cells containing a black rectangle (on) indicate that that channel will be triggered at that semiquaver position.

NOTE: For **Drums** to make any sound at all, at least one channel must have a wave file loaded, be unmuted, have its volume turned up and have some pattern entered in the sequence grid. The master volume must be turned up, the master mute unchecked and the clock running.

Bassline

Category: Signal Generators

Inputs: optional input

Outputs: mono

Bassline is a simulation of an analog monophonic synthesiser with a 16-note pattern sequencer. **Bassline** synchronises to the clock, which must be running for **Bassline** to make any sound.

Bassline also provides an audio input that can be used to feed any audio signal through its filter and envelope. When the audio input is used it replaces **Bassline's** internal oscillator, as a result neither **waveform select** or the pitch cells of the **Sequence Grid** will have any effect on sound.

Parameters

Mute

When checked **Mute** disables sound output.

Sawtooth / Square waveform select

The waveform selector buttons allow the selection of either a sawtooth or squarewave oscillator.

Volume

Volume controls the output level of **Bassline**.

Cutoff

Cutoff controls the cutoff frequency of the resonant filter.

Resonance

Resonance controls the resonance of the filter.

Env mod

Env mod (Envelope modulation) controls the amount by which the AR envelope modulates the filter cutoff frequency.

Accent

Accent controls the depth of accent applied to those notes accented in the sequence grid. Accenting is created by a combination of increased note amplitude and filter cutoff.

Decay

Decay controls the release length of the envelope. Higher values yield longer delays which is often perceived as a higher filter cutoff frequency.

Overdrive

Overdrive overdrives the filter input resulting in a distorted, raspy sounding filter; this is especially evident when large amounts of resonance are used.

LFO Rate

A sinusoidal LFO (Low frequency oscillator) is available to modulate the Env mod parameter. **LFO Rate** controls the rate (speed) of the LFO.

LFO Depth

LFO Depth controls the modulation range of the LFO. The maximum modulation range is the distance between the current Env mod value and its maximum. Reduction of LFO Depth reduces the top bound of this range. At its minimum setting no LFO modulation is applied.

Sequence Grid

The **sequence grid** represents the pattern being played. There are sixteen columns, each of which represents the state of a sixteenth note within a single bar loop. Beat numbers (1 to 4) are marked along the top of the pattern grid.

- The top row indicates the on/off state of each note in the pattern. Notes may be toggled on and off by clicking on cells in the top row.
- The second row indicates whether a note is accented or not (accent amount is controlled by the Accent knob). Accents are toggled on and off by clicking the appropriate cell.
- The third row indicates whether a note is tied on to the following note. Ties are toggled on and off by clicking on the appropriate cell.
- The final row shows the pitch of each note. This may be edited by left clicking which will display a piano keyboard where the note may be changed.

Reframe

The **Reframe** buttons rotate the whole pattern left or right by one semiquaver.

Randomize

The **Randomize** button will randomly generate a new sequence.

Mulching with Bassline

As with all the contraptions contained within **AudioMulch**, **Bassline** has a range of potential uses only limited by the ingenuity of the user. Most obviously, it functions as both a random and compositional bassline editor suitable for a wide range of popular electronic music styles. By utilising the unit's audio input function the contraption is extended to make us of a range of filters capable of transforming incoming sound signals (from stored wave files). Of particular interest is the ability of the **Bassline's Sequence Grid** to superimpose new rhythmic structures onto existing sound files.

10Harmonics

Category: Signal Generators

Inputs: none

Outputs: mono

Using the musical concept of Additive synthesis as its theoretical starting point, **10Harmonics** is a ten harmonic additive signal generator. Additive synthesis has its premise in the notion that complex musical sounds can be assembled by adding together a number of relatively simple sound components (which themselves contribute to the overall quality of the produced sound rather than functioning as individually identifiable sounds). In the case of **10Harmonics**, the first ten harmonics of a selected fundamental frequency can be added in varying amounts to the original tone to produce a sound with the desired harmonic content. **10Harmonics'** output is the sum of 10 sinusoidal oscillators spaced at integer multiples of the fundamental frequency. The fundamental frequency and the amplitudes of each harmonic can be individually controlled. A display of the output waveform is also provided.

Parameters

Frequency

The **Frequency** parameter controls the fundamental frequency (f_1) of the additive synthesis. The frequency may be typed into the edit box provided or adjusted using the slider. Frequency ranges from 20 to 1000 Hz.

Gain

The **gain** parameter controls the overall output volume of the contraption.

Harmonic Amplitudes

Individual sliders are provided to control the amplitude of each harmonic. The amplitude sliders indicate the absolute gain of each harmonic, while the waveform display shows a normalized view of the composite waveform.

BubbleBlower

Category: Signal Generators

Inputs: none

Outputs: left, right

Under Construction

BubbleBlower is a stored sample granulator based on the CloudGenerator program by Curtis Roads and John Alexander. **BubbleBlower** is in many ways similar to DLGranulator, but differs in two main respects: Firstly, the distribution of grains in time is based on a Density factor per unit time as opposed to DLGranulators interonset time parameter. Secondly, BubbleBlower draws sound for the grains from a sample loaded from a file rather than from a delay line filled in real-time.

Parameters

File

Specifies the soundfile to use as the source for granulation.

Amp - Grain Amplitudes

Specifies the range of possible amplitudes available for each grain.

Pan - Grain Pans

Specifies the range of possible stereo panning locations available for each grain.

Inskip - Sample inskip

Specifies the range of possible sampling inskips available for each grain. Each grain is individually sampled from stored sample loaded from the file specified by **File**.

Trans - transposition factor

Specifies the range of possible transposition factors available for each grain. The notch marks unity, the slider has a range of +/- 2 octaves. Transposition factor effects the rate at which each grain is played back. Positive transposition factors will have the effect of shifting the output higher in pitch while negative factors will lower the pitch of the output.

Density

Specifies the average density of grains, expressed in number of grains per second.

Max Grains - Limit maximum simultaneous grains

Due to the limited processing power of computers it is not realistic to mix an infinite number of overlapping grains in real time. The maximum allowed with BubbleBlower is 20, this may be too many for slower systems to mix in real time. Max Grains is provided to avoid audio glitches on slower machines. Lowering Max Grains will lower the CPU load but will 'thin out' granulations using a lot of overlapping grains.

Quant - quantization amount and quantization grid

When the clock is running, BubbleBlower allows the onset times of all grains to be quantized. The slider allows the amount of quantization to be set, ranging from none (minimum) to total (maximum). The combo box allows selection from a variety of quantization pulses. Quantization is only active when the clock is running.

GDur - Grain Duration

Specifies the range of possible durations available for each grain. Gdur can range from 10 to 500 milliseconds.

Shape - Envelope shape

Each grain has an amplitude envelope consisting of an attack, sustain and decay portion. TrapFact determines the duration of the sustain portion relative to the duration of the attack and decay portions. When shape is 0, the envelope is a triangle, when it is 1 the envelope is a rectangle.

Skew - Envelope Skew

Specifies the range of possible envelope skew factors available for each grain. Skew adjusts the relative duration of the attack and decay portions of the grain envelope. Smaller values of skew lessen the attack time and increase the decay time, larger values of skew lessen the decay time and increase the attack time. Both extremas of skew are useful for creating more interesting grain profiles when longer grain durations are being used.

LoopPlayer

Category: Signal Generators

Inputs: none

Outputs: left, right

LoopPlayer is a clock synchronised sample loop player. It is commonly used for playing back sampled or pre-recorded drum loops in time with other clock-synchronised contraptions. **LoopPlayer** will not output any sound if the clock is not running. The whole wave file is loaded in to ram, so be careful using large files on machines with small amounts of physical ram. The controls for **LoopPlayer** are described below.

Parameters

File

The **file** panel identifies the wave file currently loaded into **LoopPlayer**. Files may be selected by clicking on the open button next to the file panel. An exclamation mark (!) next to the file name indicates that the file does not exist. While mono and stereo files recorded at varying sample rates can be opened using **LoopPlayer**, it is important to note that the contraption will store and play no more than the first fifteen seconds of any file.

Mute

When checked, **Mute** silences the output of **LoopPlayer** from the start of the next bar. When mute is unchecked, **LoopPlayer** resumes audio output from the start of the next bar.

Bars

Bars is the number of bars contained within the sample loop. It is used to determine how often to re-trigger the sample, and when stretch is active – how to transpose the sample to conform to the current tempo.

Phase

Phase shuffles the sample forward or backwards relative to the clock in semiquaver increments.

Stretch

Stretch transposes the sample so that its playback duration exactly matches the duration of the number of bars specified with **Bars**, at the current tempo. With **Stretch** checked, the sample will remain synchronised to the clock even if the tempo is altered. The pitch of the sample will be altered however.

Mulching with LoopPlayer

While **LoopPlayer's** role within most AudioMulch documents could be considered a secondary one it should not be ignored as a useful means of assembling sound fragments and samples into ordered wholes. For example: the average Pentium machine running Windows 95 can support 12 **LoopPlayers** running simultaneously. Theoretically, these twelve samples (if divided into drum loops, basslines, musical passages, and vocal phrases) could then be re-constructed in realtime to form live and evolving remixes of existing material. Using the **mute** and **stretch** functions, the user can then both alter and alternate

between those samples loaded

NastyReverb

Category: Effects

Inputs: mono

Outputs: left, right

As the name suggests, **NastyReverb** is a very ordinary computer-music circa 1980 style reverb.

Parameters

Reverb time

The **Reverb time** parameter controls the time the reverb takes to decay to silence, **Reverb time** ranges from 100ms to 10 seconds.

Wet/dry mix

Wet/dry mix controls the ratio of input vs. reverberated signal.

Mulching with NastyReverb

An everyday concept, Reverb is the naturally occurring process of sound reflection. When we listen to a sound we hear not only the original signal, as it travels from source to ears, but also a series of later reflections as the signal bounces from source to wall to ears. These reflections can take a variety of sonic paths arriving at varying intervals and strengths thus creating the reverb effect. Used subtly, NastyReverb can create the illusion of space when applied to an otherwise flat mono input.

Flanger

Category: Effects

Inputs: mono

Outputs: mono

Originally produced using two slightly out of sync reel-to-reel tape machines, the flanger imposes a modulated whooshing sound (phase cancellation) across the input signal. In this digital implementation, a comb filter frequency modulated by a sine wave is used.

Parameters

Range

Range specifies the minimum and maximum frequencies between which the filter is swept. The extremes of range are 20hz and 4000hz. Range is controlled using a *Range Slider*; the use of which is described in the Using Contraption Controls section of this help file.

Rate

Rate specifies the rate at which the filter is swept up and down between its minimum and maximum frequencies. The rate ranges from 1 cycle every 100 seconds to 100 cycles per second.

Feedback

Feedback controls the amount that the flanger resonates at its cutoff frequency, higher values of feedback produce a more noticeably pitched effect.

Wet/Dry mix

Wet/dry mix controls the ratio of input vs. flanged signal.

Phaser

Category: Effects

Inputs: mono

Outputs: mono

Under Construction

DLGranulator

Category: Effects

Inputs: mono

Outputs: left, right

DLGranulator is an implementation of a delay line granulator. A delay line granulator samples small sonic fragments (typically < 100ms) from a delay line, and reassembles them into a stream of enveloped "grains". Granulators are useful for generating dense textures, pitch shifting and other audio mulching tasks. A "grain" refers to a single sampled fragment with an envelope applied to it. Each grain has parameters that determine how the grain is sampled from the delay line and how it is enveloped and panned. The *interonset time* parameter (IOT) determines the time between the start of one grain and the start of the next in the output stream. Other parameters control input, output, feedback and wet/dry mix levels. Many parameters make use of *range sliders* (see Using Contraption Controls) to specify a range of values; in such cases each grain is assigned a random value from within the specified range.

Parameters

InGain - Input Gain

Specifies the amount the input signal is scaled before it is granulated.

Amp - Grain Amplitudes

Specifies the range of possible amplitudes available for each grain.

Pan - Grain Pans

Specifies the range of possible stereo panning locations available for each grain.

Delay - Grain Sampling Delay

Specifies the range of possible sampling delays times available for each grain. Each grain is individually sampled from the delay line. If the minimum and maximum values of Delay are the same, the output will be a granulated version of the input signal, delayed by the amount specified. If the minimum and maximum values of Delay specify a range, this will have the effect of time smearing the input signal (each grain will select a random delay time from within the specified range). Delay ranges from 0 to 9.5 seconds.

Freeze - Delay line freeze

Freeze will pause input to the delay line. This allows the delay line to be statically sampled for time freezing effects.

Feed - Feedback

Specifies how much of the granulated output is fed back into the delay line input. This can be used to create arpeggiating effects when the transposition factor is not unity.

Mix - Wet / dry mix

Specifies the ratio between granulated and input sound presented at the output.

Trans - transposition factor

Specifies the range of possible transposition factors available for each grain. The notch marks unity, the slider has a range of +/- 2 octaves. Transposition factor effects the rate at

which each grain is played back. Positive transposition factors will have the effect of shifting the output higher in pitch while negative factors will lower the pitch of the output.

IOT - interonset time

Grains are mixed into the output stream in an overlapping sequence, interonset time determines the time from the beginning of one grain to the beginning of the next. If the grain duration (GDur) is less than the interonset time, a particled texture will result. When grain durations exceed interonset time, grains will overlap making it possible to create smooth textures. Interonset time ranges from 5ms to 2 seconds.

Max Grains - Limit maximum simultaneous grains

Due to the limited processing power of computers it is not realistic to mix an infinite number of overlapping grains in real time. The maximum allowed with DLGranulator is 20, this may be too many for slower systems to mix in real time. Max Grains is provided to avoid audio glitches on slower machines. Lowering Max Grains will lower the CPU load but will 'thin out' granulations using a lot of overlapping grains.

Quant - quantization amount and quantization grid

When the clock is running, DLGranulator allows the onset times of all grains to be quantized. The slider allows the amount of quantization to be set, ranging from none (minimum) to total (maximum). The combo box allows selection from a variety of quantization pulses. Quantization is only active when the clock is running.

GDur - Grain Duration

Specifies the range of possible durations available for each grain. Gdur can range from 10 to 500 milliseconds.

Shape - Envelope shape

Each grain has an amplitude envelope consisting of an attack, sustain and decay portion. TrapFact determines the duration of the sustain portion relative to the duration of the attack and decay portions. When shape is 0, the envelope is a triangle, when it is 1 the envelope is a rectangle.

Skew - Envelope Skew

Specifies the range of possible envelope skew factors available for each grain. Skew adjusts the relative duration of the attack and decay portions of the grain envelope. Smaller values of skew lessen the attack time and increase the decay time, larger values of skew lessen the decay time and increase the attack time. Both extremas of skew are useful for creating more interesting grain profiles when longer grain durations are being used.

RingAM

Category: Effects

Inputs: carrier, optional modulator

Outputs: mono

The operation of the **RingAM** is based on the processes of Ring and Amplitude modulation in which an external input (carrier input signal) is combined with either an internal sinusoidal modulator or a second external input – serving as a modulator. The process at its most basic level involves the combination of two discrete signals to produce a single output, which is comprised of both the sum of and difference between those original frequencies. These new frequencies are called sidebands, the upper sideband relating to the sum of the frequencies and the lower relating to their difference.

RingAM facilitates three different versions of this modulation process. Amplitude modulation (AM), Ring modulation (Ring Mod), and External modulation. Both **AM** and **Ring Mod** utilise an internal sinusoidal modulator (*AM multiplying the input signal by a positive sinusoid ranging in amplitude from 0.0 to 1.0 while Ring Mod uses an a/c sinusoid ranging from -1.0 to 1.0*). When an external modulator is provided at the second input, the internal sinusoidal modulator is disabled, hence the **Frequency** and **Ring Mod / AM** controls will have no effect. External modulation is always full cycle (a/c) ring modulation.

Parameters

Frequency

Frequency controls the frequency of the modulator. The available range is 20Hz to 1500Hz.

Ring Mod / AM

The **Ring Mod** and **AM** radio buttons switch between ring modulation and amplitude modulation.

Mulching with RingAM

Compared to many of the other effects featured within AudioMulch (e.g. Reverb, Flanger, Phaser), Ring and Amplitude modulation are neither widely known or used within mainstream musical circles. This lack of use can be attributed to the non-harmonic and dissonant sounds produced by the operation. Having little in common with the harmonic qualities of the original input, the harsh product of the RingAM may not be to everyone's taste. This dissonance can be smoothed, however, by mixing it back in with the original input. By running the output of the RingAm and the original signal through a mixer this balance can be adjusted to create an unusual alternative harmony rather than a seemingly irrelevant tone.

SDelay

Category: Effects

Inputs: mono

Outputs: left, right

As its name suggests a delay takes an input signal and replays it after a defined delay time. This delayed signal can then be mixed back in with the original signal to create an echo effect. By rechannelling the delayed signal back through the delay line (**Feedback**) the number of repetitions can be increased. The **Feedback** value determines the number of repetitions or "decay time" of the delay. **SDelay** is a stereo ping-pong delay line. It is configured with its right output fed back into both inputs to create stereo bouncing type delay effects. The delay times for the left and right channels may be independently varied, either in milliseconds or semiquavers relative to the tempo.

Parameters

Delay Units

Delay Units allows delay times to be specified either in milliseconds or semiquavers.

Delay times

Separate **delay times** for the left and right channels may be specified either by typing delay times directly into the edit boxes provided, or by using the increment/decrement arrows next to the edit boxes. Delay times can range from 20ms to 2 seconds

Feedback

Feedback controls the amount of delayed signal from the right output which is fed back into the input. Larger values of feedback cause the delays to echo for longer. Beware of feedback values close to maximum, these may cause the delay line to 'blow up'.

Wet/Dry mix

Wet/dry mix controls the ratio of input vs. delayed signals.

Mulching with SDelay

Put simply, a delay effect is useful for filling out or widening an instrument's sound. This widening can serve a range of purposes with differing contexts demanding varied approaches. As a part of the ever-changing network of contraptions facilitated by the structure of AudioMulch these differing applications are particularly relevant.

When combined with **Drums**, **SDelay** can either serve to "swing" the programmed sequence or may be used to increase the complexity of a relatively simple rhythmic pattern. Using a relatively small **Delay time**, equivalent to or less than a semiquaver (125ms), with no **Feedback** and the **Wet/Dry mix** set at maximum – Wet – a simple pattern would be shifted off the beat in relation to other clock synched signals creating a swing feel. Multiple **Drums**, with or without delay, could be used in this way if only certain aspects of the pattern are to be "swung" (ie. the snare). By increasing **Feedback** and mixing in the original source (drying out the mix) the feel is changed. This function is particularly useful when trying to create more complex rhythms as **Drums** offers only a basic sequencing

pattern.

By running live vocals or instruments through a **SDelay** with a **Delay time** of between 50-100ms and an evenly balanced mix, it is possible to create a doubling effect which recreates the sound of another vocalist/instrumentalist singing or playing in unison.

SSpat

Category: Effects

Inputs: mono

Outputs: left, right, reverb send

SSpat is a stereo spatialiser, allowing the projection of a moving mono source signal into a virtual plane lying behind a pair of stereo speakers. The path along which the source travels may be specified, along with various parameters effecting the apparent dimensions of the virtual plane. A reverb send is provided to allow a more realistic room simulation to be achieved. The reverb send is a delayed version of the input, amplitude scaled relative to its distance from the rear of the virtual plane.

Parameters

Path

The **Path** pane allows the specification of an arbitrarily complex spatialisation path. The path defines the shape (but not necessarily the position or scale) of the trajectory along which the source travels. The path is specified by a looped spline, which consists of 3 or more segments. Segments are added to the path by clicking anywhere along the existing path. To move a segment, click and drag a handle (square box). To delete a segment, hold down the control key and click on the segment you wish to delete.

Trajectory

The **Trajectory** pane controls the location of the path within the virtual plane. The blue **L** and **R** boxes at the bottom of the **Trajectory** panel indicate the location of the speakers relative to the trajectory (viewed from above, with the listener below the pane). The path can be moved by dragging with the mouse, rotated by dragging up and down with the control key depressed, and scaled by dragging up and down with the shift key depressed.

Velocity

Velocity controls the rate at which the source travels around the trajectory.

Scale

Scale controls the relative size of the virtual room; this effects the amplitude and doppler shift of the source. Larger values of scale make the rear of the virtual plane further away.

Doppler Amount

Doppler Amount attenuates the amount of doppler shift. At minimum, no doppler shift is applied to the source.

Amp Amount

Amp Amount effects the amount of attenuation applied to the source as a result of its distance from the virtual listener. At minimum, no attenuation is applied to the source.

Separation

Separation effects the apparent width of the stereo image. Larger values of separation produce more pronounced left-right panning.

Reverb Scaling

Reverb Scaling works similarly to **Amp Amount** but applies to the distance based scaling of the source as it is sent to the reverb send output.

5Combs

Category: Filters

Inputs: mono

Outputs: mono

A comb filter is a really short delay line with feedback. The delay is usually so short that you can hear it resonating at a specific pitch. This occurs as the result of periodic cancellation and reinforcement, which in turn creates a series of harmonically related peaks and notches throughout the audio frequency range. Another way to think about it is as a guitar string, except instead of strumming, you feed an audio signal into it and it resonates.

5Combs is a bank of 5 comb filters in parallel. Maintaining the analogy above the 5Combs contraption is like a 5 stringed guitar, except that the frequency range of each string is changeable. By selecting and amplifying particular frequencies within the input signal one can either create a dissonant or complementary chordal drone from a signal which originally seemed to possess no harmonic quality (i.e. Drums).

Parameters

Master Gain

Master Gain adjusts the overall output volume of 5Combs

For each of the 5 filters:

Frequency

Frequency controls the fundamental frequency of each comb filter. Frequency can be specified with the slider, or entered as a musical note on a piano keyboard by clicking on the note name next to the slider.

Decay

Decay determines how long each filter "rings" for. Depending on the input signal your mileage may vary. The decay slider theoretically ranges from almost 0 to 10 seconds.

Gain

Gain provides independent gain of each comb filter.

ParaEQ

Category: Filters

Inputs: mono or left, right

Outputs: mono or left, right

MParaEQ and **SParaEQ** are mono and stereo versions respectively of a parametric equalizer consisting of a high shelving filter, two sweepable midrange boost/cut filters and a low shelving filter. The ranges of each filter section is described below:

	Cutoff/Center frequency (cf)	Bandwidth (bw)	Gain
Hi	3 – 18Khz	-	+/- 30db
Mid1	50Hz – 15Khz	10Hz – 2KHz	+/- 30db
Mid2	50Hz – 15Khz	10Hz – 2KHz	+/- 30db
Low	20Hz – 3Khz	-	+/- 30db

S*Mixer

Category: Mixers

Inputs: 1 (left), 2 (right), 3 (left), 4 (right), . . .

Outputs: left, right

S*Mixers (where * indicates the number of stereo input pairs), are stereo mixers with a master gain control (**Master**) and individual gain controls for each stereo input pair (**1&2**, **3&4**, etc.). Mono signals applied to either input of a stereo pair are automatically bridged to the other input of that pair.

P*Mixer

Category: Mixers

Inputs: 1 (mono), 2 (mono), . . .

Outputs: left, right

P*Mixers (where * indicates the number inputs), are mixers with mono inputs and a stereo output. A master gain control (**Master**) and individual pan and gain controls for each input (**1, 2**, etc.) are provided.

M*Mixer

Category: Mixers

Inputs: 1 (mono), 2 (mono), . . .

Outputs: mono

M*Mixers (where * indicates the number inputs), are mono mixers with a master gain control (**M**) and individual gain controls for each input (**1, 2**, etc.).

Crossfader

Category: Mixers

Inputs: 1 (left), 2 (right), 3 (left), 4 (right)

Outputs: left, right

Crossfader is designed to crossfade between two stereo input signals. Mono signals applied to either input of either stereo pair are automatically bridged to the other input of that pair. As well as the crossfade slider, master gain (**Master**), and individual trim knobs (**1&2** and **3&4**) for each stereo pair are provided.

Matrix

Category: Mixers

Inputs: 4 or 8 matrix input

Outputs: 4 or 8 matrix outputs

Under Construction

4x4Matrix and **8x8Matrix** provide a means for dynamically routing signals without repatching the document. The grid represents each input as a row, and each output as a column. Enabling cells (by clicking on them, or navigating with the cursor keys and selecting with the space bar) causes a connection to be made between the input of the specific row and the output represented by that column. More than one input may be connected to an output at the same time, in which case the signals are summed.

The **FadeTime** parameter determines the time taken for a newly enabled input to be faded into the output, or a newly disabled input to be faded from the output. **FadeTime** can range from 0ms to 100seconds. Short values of fade time can be used to prevent clicks when switching, longer fade times can be used to create slowly fading mixes.

S*Buss

Category: Busses

Inputs: 1 (left), 2 (right), 3 (left), 4 (right), . . .

Outputs: left, right

S*Busses (where * indicates the number of stereo input pairs), are stereo summing busses. No property editor is provided for **S*Busses**. All signals are mixed to the output with unity gain. Mono signals applied to either input of a stereo pair are automatically bridged to the other input of that pair.

M*Buss

Category: Busses

Inputs: 1 (mono), 2 (mono), . . .

Outputs: left, right

M*Busses (where * indicates the number of inputs), are mono summing busses. No property editor is provided for **M*Busses**. All signals are mixed to the output with unity gain.

VST Plugins

Category: VST Plugins

Inputs: dependent on the particular plugin

Outputs: as with inputs

Using **VST Plugins** you can expand AudioMulch's signal processing capabilities using effects designed by other people for the VST plugin architecture. There are currently over 50 free and shareware plugins available on the internet, check the AudioMulch website (<http://www.audiomulch.com>) for links to some relevant sites.

When AudioMulch is installed it will create the *VSTPlugins* directory so that all users have to do is place their downloaded plugins into this directory. The VST Plugins submenu of the new contraptions menu reflects the structure of the menu structure of the VSTPlugins directory (i.e. folders within the VSTPlugins directory appear as submenus in the VST Plugins menu). This construction allows plugins to be collected in similar useful groupings through the nesting of folders. This is useful for organising a large number of plugins according to type or creator for example.

VST Plugins can be used just like other contraptions. Some plugins have their own graphical user interface that will appear as their property editor. For other plugins AudioMulch provides a default editor displaying all parameters and their values. To select a parameter for editing click on it and use the slider to set the value of that parameter.

DigiGrunge

Category: Beta

Inputs: 1

Outputs: 1

DigiGrunge allows the controlled application of two major types of digital distortion: bit depth quantisation noise and sample rate decimation aliasing. It can be used to demonstrate digital distortion and to create extremely clipped and distorted sounds characteristic of low bit rate sampling techniques.

As the bit depth quantisation technique employed is dependent on input level, the **Input Gain** control allows the input signal to be adjusted to obtain an optimal amount of quantisation noise.

The **Bit Depth** parameter controls the number of effective bits being used to represent the signal. A Bit depth of 1 reduces the signal to either on or off (plus a sign bit) creating an extreme clicking distortion. Such a signal can be used as an input for resonant contraptions like 5Combs or Phaser to create sounds with a percussive quality. Less extreme settings of Bit Depth are useful for creating the type of digital noise commonly associated with older digital samplers and computers.

The **Decimation** parameter controls the amount of effective sample rate reduction. A decimation setting of 1 maintains the signal at its original sampling rate, a decimation setting of 2 reduces the effective sample rate to half the original - usually from 44.1kHz to 22.05kHz. Note that this effect is not intended to produce clean down sampling - it introduces a large amount of digital distortion by sampling and holding the input signal at the effective sampling rate.

An effect called digital aliasing causes all frequencies in the input signal that are above half the effective sample rate (technically referred to as the Niquist frequency) to be reflected below the Niquist frequency by the same amount as they were above it. Thus as the decimation parameter is increased frequencies a long way above the Niquist frequency get reflected progressively lower below it, often resulting in low frequency tones and other artifacts.

Shaper

Category: Beta

Inputs: 1

Outputs: 1

Shaper creates harmonic distortion using a Chebychev polynomial waveshaper. This distortion method allows the user to graphically edit the relative strength of the first 27 harmonics that would be induced given a full strength sine wave input signal. In practice various signals may be passed through **Shaper**, the harmonic content of the resultant distorted signal will be related to the user specified shaping function and the harmonic content and amplitude of the input signal.

One use of **Shaper** is as a flexible distortion processor. Often only distortion of the first few harmonics is required to create complex distortion effects. The more complex the input is, the less distortion is required to create a noticeable effect.

Shaper may also be used for non-linear waveshaping synthesis. By passing a full strength sine wave or other simple waveform (such as those generated by TestGen and Additive10) through **Shaper** and modulating the **Input Gain**, various harmonic spectra may be synthesized. When a full strength sine wave is passed through Shaper with maximum **Input Gain**, the shaping function introduces distortion such that the output signal contains harmonics whose relative strengths are those specified in the **Harmonic weightings** graph.

Shaper is very sensitive to input level - varying amounts of distortion will be produced depending on the level of the input. The harmonic content introduced by a less than full strength sine wave or non-sinusoidal input is not easily predicted. By varying the **Input Gain** interesting shifts in the harmonic balance may be created. Overdriving Shaper may produce harsh clipping distortion, care should be taken when setting the **Input Gain** parameter when such clipping is undesirable.

Parameters

Input Gain controls the level of the signal entering the shaping function - owing to the internal scaling algorithm, Input Gain effects only the amount of shaping distortion applied, and not the output level of Shaper.

The **Output Gain** parameter varies the output level of Shaper independent of the Input Gain.

The **Harmonic weightings** graph allows the relative strength of the harmonics introduced by the shaping function to be specified. As the mouse is moved over the graph, individual harmonics are highlighted - clicking on a harmonic and dragging up and down sets the relative strength of the selected harmonic. Odd harmonics are displayed in red, even harmonics are displayed in blue.

By holding down the shift key and clicking and dragging the mouse horizontally it is possible to 'paint' all of the Odd harmonics. The same task can be performed for even harmonics by pressing the control key while dragging. When both the shift and control keys are depressed all harmonics can be "painted".

Harmonic 0 is equivalent to the input signal - if only this harmonic is specified in the Harmonic Weightings graph Shaper will output the input signal with amplitudes up to the maximum input level - exceeding this level will introduce audible clipping distortion which can be eliminated by reducing the **Input Gain**.

The **Shaping function** displays a graph of the distorting function that is applied to the input signal to produce the output signal. The input is graphed on the horizontal axis, output on the vertical axis.

Invert

Category: Beta

Inputs: 1

Outputs: 1

Invert is a simple contraption that inverts the polarity of its input signal - this is sometimes referred to as "phase inversion" or a 180 degree phase shift. A simple pseudo stereo effect may be created by panning a mono signal to one speaker and an inverted version to the other speaker. Invert may also be useful for implementing some phase related encoding and decoding schemes.

***Gain**

Category: Beta

Inputs: 1, 2, 4, and 8.

Outputs: 1, 2, 4, and 8.

MGain, **SGain**, **QGain** and **OGain** are one, two, four and eight input/output gain modules that attenuate each of their inputs by a gain factor controlled by a single knob. A Gain contraption may be used to apply global attenuation to a set of signals that are mixed using AudioMulch's automatic bussing feature (multiple outputs patched into a single input).

Overview of AudioMulch Tutorials

(Tutorials Under Construction)

As has been suggested above, the use and application of AudioMulch can be limited only by the technical attributes of your machine and the depth and scope of your own imagination. One cannot however imagine the result of a complex network of sound synthesis and processing contraptions if there does not exist a reasonable understanding of the varied capabilities and combined potentials of the program itself. So far this help file has introduced you in some detail to the workings of the individual contraptions (Contraption reference) and has provided opportunities to experience the basic capabilities of the software (example files). The following section, focuses more on the specific knowledge required to perform a number of task and genre based operations.

Designed as a computer music tool, AudioMulch lends itself to both unusual and what for the lack of a better expression we will refer to here as commercial uses. It can be used in live performance situations or as processing tool for the composition of recorded pieces. The following sub-sections offer a range of these applications; from basic use to specific genre based workshops. Importantly however, AudioMulch applies no limits to the combination of its many parts, so there are also several sections included here which focus on particular contraptions. Finally, it should be said that these tutorials by no means provide an exhaustive account of the many uses of the software and it would be just as relevant to provide "Big Beats" or "Vocal/Instrument processing" workshops here – maybe later!

A Beginners Tutorial

The following tutorial demonstrates the construction of a basic AudioMulch document and will hopefully get you started on your exploration of the many elements of the software.

Before you undertake this tutorial it is recommended that you familiarise yourself with the major user interface elements of AudioMulch as described in the *Using AudioMulch* section of this help file.

- Right-click on the Patcher pane and choose **New->IO->SoundOut** from the context menu. The **SoundOut** contraption will appear on the Patcher pane.
- Right-click on the Patcher pane and choose **New->SignalGenerators->Bassline**. The **Bassline** is a simple monophonic synthesiser.
- Drag a line from the output of the Bassline to the leftmost input of the SoundOut.
- Double-click on the Bassline contraption. The Property Editor for the Bassline will appear in the Property pane.
- Press the Play button in the toolbar to start the clock running. This will automatically turn on the sound output as well.
- Slowly increase the Volume control on the Bassline Property Editor until you can hear the sound.

You should now be hearing the default repetitive pattern of the bassline synthesiser. Your patch should look like this:



On the Bassline property editor, there is a grid, which represents the pattern played. There are sixteen columns, each of which represents the state of a sixteenth note within a single bar loop. Beat numbers (1 to 4) are marked along the top of the pattern grid.

The top row indicates the On/Off state of each note in the pattern

The second row indicates whether a note is accented or not (accent amount is controlled by the Accent knob).

The third row indicates whether a note is tied on to the following note.

The final row shows the pitch of each note. This may be edited by left clicking which will display a piano keyboard where the note may be changed.

- Leave the notes switched on for each of the beats marked along the top of the grid, (1, 2, 3 and 4), but switch off all the in-between notes. Then switch on the note next to beat 1.

You should have a pattern which looks like this:

—

- Right-click on the Patcher pane and choose **New->Effects->SDelay**. We will now patch the Stereo Delay in between the Bassline and the SoundOut.
- Click on the line that links the Bassline to the SoundOut and press the delete key. This will clear the connection between the two contraptions.
- Connect the output of the Bassline to the input of the Stereo Delay.
- Connect the outputs of the Stereo Delay to the corresponding inputs of the SoundOut.
- Double click on the StereoDelay contraption. This will open its Property Editor in the Property pane.
- Change the Left value of the delay to 375. Change the Right value to 625.
- On the bottom of the grid in the Bassline contraption, there is a row of note values. By default, these values are set to c3 (or middle C in musical terms).
- Click on the first c3. A keyboard will pop up. Use the keyboard to choose the same note one octave up. When you do so, the note should read c4 instead of c3. You will hear the note in the resulting pattern.

Your patch should look like this:



You can click and drag the contraptions around in the Patcher Pane if you need to make more room.

We will now insert a Flanger into the line between the Bassline and the StereoDelay.

- Right-click on the Patcher pane and choose **New->Effects->Flanger**.
- Click on the patch cord between the Bassline and the Stereo Delay and delete it. The sound will stop.
- Patch the output of the Bassline into the input of the Flanger.
- Patch the output of the Flanger into the StereoDelay. You will hear the effect of the Flanger immediately.
- Double-click on the Flanger to open it's Property Editor.
- Try experimenting with different Flanger settings to find a sound you like.

Your patch should look something like this:



Try changing the Cutoff, Resonance and Env mod parameters of the Bassline to find some interesting sounds. Try changing the Wet/Dry mix on the StereoDelay contraption to hear more of the original Bassline input against the delayed output.

Further Reading

The following is a list of books and articles covers various aspects of making music with computers. Most are concerned with the manipulation of sound rather than more traditional 'electronic music' applications. The AudioMulch web site contains a list of online resources that may also be useful.

Moore, F.R. 1990. "Elements of Computer Music," Prentice Hall, New Jersey.

Roads, C. 1996. "The Computer Music Tutorial," MIT Press, Massachusetts.

Wishart, T. 1994. "Audible Design," Orpheus and the Pantomime Ltd. York.

Chadabe, J. 1997. "Electric Sound: The Past and Promise of Electronic Music," Prentice Hall, New Jersey.

Command Line Parameters

When AudioMulch is launched from a dos box, it is possible to specify extra arguments to effect AudioMulch's behavior. These arguments are described below.

Usage:

Mulch.exe

Or

Mulch.exe *documentName*

Or

Mulch.exe [*options*] *documentName*

Where *options* can include any of the following:

/d Enable real-time audio

/c Start clock

Menu Item Reference

File

New	Create a new, empty AudioMulch document
Open. . .	Open an existing AudioMulch document
Reopen	Open a recently used document
Revert	Abandon all edits since last save
Save	Save the current document
Save As. . .	Save the current document under a different name
Export as Wave File...	Save an audio segment as a wave file
Exit	Exit AudioMulch

Edit

Undo	<i>not implemented</i>
Redo	<i>not implemented</i>
Cut	Cut current selection to clipboard
Copy	Copy current selection to clipboard
Paste	Paste clipboard data
Clear	Clear the current selection
Select All	Select all
Settings...	Adjust device settings and paths for files

View

Toolbars	Show or hide the various toolbars
Status Bar	Show or hide the status bar
Patcher Pane	Show or hide the patcher
Properties Pane	Show or hide the properties pane
Volume Control	Display the soundcard's volume control window

Control

Enable Audio	Enable or disable real-time audio
Enable MIDI Controllers	Enable or disable real-time MIDI parameter control
Parameter Modulation...	Configure MIDI parameter control
Play	Start the clock
Stop	Stop the clock
Loop	<i>not implemented</i>
Chase sync	Enable synchronisation with a MIDI clock source
Generate sync	<i>not implemented</i>

Items in the Control menu are disabled if the required contraptions are not present in the document.

Help

Register AudioMulch	Displays shareware registration information
Help Topics	Displays this Help File
About AudioMulch	Displays version and Copyright information

Export as Wave File

When "Export as Wave File..." is selected from the file menu the Export as Wave File Dialog appears. A number of options are presented with regard to the duration of the particular file segment to be save and the method by which it is generated. Once the appropriate options have been configured click the Export button to generate the output file, or click the Cancel button to abort the operation.

Three export modes are provided:

Clock Synchronised Pattern creates a file beginning exactly with its duration specified in beats (measured using the current tempo). The first beat of the first exported bar begins exactly at the start of the file. Sounds exported in this way will loop exactly from beginning to end, and may be easily used in contraptions such as LoopPlayer.

Timed Segment allows the duration of the exported segment to be specified in seconds.

Processed SoundIn File requires that the SoundIn contraption is being used as the primary signal generator and that it is configured to play an input file. When this option is used the input file is played from the beginning to enable the easy generation of a processed version of the SoundIn file.

The **Record** parameter is used to select the number of bars (or seconds) you wish to record. The **Pre-roll** parameter determines how many bars (or seconds) will play before the actual recording starts. This parameter lets you ensure that all sounds have started and that delay times are synchronised. It's usually a good idea to allow at least 2 bars pre-roll before recording.

Settings Dialog

The Settings dialog box is accessed from the Settings... item in the Edit menu or by pressing the F4 function key. The Settings dialog box provides a series of tabs for configuring input and output sound card settings, MIDI device and synchronisation settings, and the directories AudioMulch searches to locate wave files used in Contraptions such as Drums and LoopPlayer.

The Audio Output tab

Devices

The Audio Output tab contains all settings pertaining to AudioMulch's real-time audio output capabilities. The Devices grid indicates which soundcard devices are used by which input devices (SoundIn and AuxOut1 - 7). By clicking on a row of the grid a combo box is provided for selecting a device. At present only devices providing 44k stereo output are supported. On many systems only a single sound output device will be available – in such cases this should be used as the SoundOut device and the device settings for AuxOut – 7 can be set to "(None)". The following three paragraphs discuss configuring AudioMulch for multichannel output and may be skipped if a stereo sound card is being used.

AudioMulch supports up to 16 channels of real-time audio output using 8-stereo pairs provided by the SoundOut and AuxOut1 – 7 contraptions. The Devices grid allows you to select which device is used by each output contraption.

Manufacturers provide support for multi-channel soundcards using two different methods: a group of stereo devices, or a single multi-channel device. In the former case there can be a one to one correspondence between AudioMulch output contraptions and soundcard devices, as both are stereo. AudioMulch supports the case of a single multi-channel device by assigning pairs of device outputs sequentially to the contraptions that are assigned that device. When using a multi-channel device the output pair used by a particular contraption is indicated in brackets next to the device. For example "WaveCenter ADAT 1:8 (5:6)" would indicate that the fifth and sixth outputs of the "WaveCenter ADAT 1:8" device (an eight-channel device) are assigned to that contraption.

More than one contraption may use the same device, or the same stereo pair of a multi-channel device, in these cases the contraption outputs will be mixed to the soundcard outputs. When a stereo pair is used by more than one contraption an asterisk (*) will appear next to all instances of that device.

Buffer Size and Number of Buffers

To provide seamless audio output, AudioMulch continuously streams a number of buffers to the output device(s). The number and size of these buffers can be changed using the Buffer Size and Number of Buffers combo boxes. Depending on the speed of your system and the quality of your sound card and drivers, different values for the Buffer Size and Number of Buffers settings will be optimal. Settings may be changed while AudioMulch is generating sound by clicking the Apply button or closing the Settings dialog box using the OK button. By default AudioMulch uses four 16384 byte buffers.

The total duration of the buffers, indicated as total latency next to the Number of Buffers combo box, determines the time from when Contraption parameters are edited on screen to when the change is reflected by the audio output. This may also effect the smoothness of

continuous parameter updates.

If desired you can experiment to determine the lowest working values on your system by reducing both buffer size and number of buffers until the audio output begins to break up with clicks and gaps. When using contraptions that continuously read or write to the hard disk, such as SoundIn and SoundOut, larger buffer sizes will probably be more appropriate than in cases where no disk activity is required.

Overdrive

The overdrive checkbox allows you to trade off performance for stability under marginal conditions – when checked AudioMulch will give creating stable real-time output priority even when the CPU load is high (approaching or beyond 100%). When audio production is given high priority the user interface will become sluggish and may even freeze. Unchecking the overdrive checkbox will result in less stable audio output when the CPU load is high, but guarantees that the program will not freeze.

It is suggested that Overdrive be disabled except when using documents that are known not to overload the computer (ie those that don't exceed 85% CPU Load).

The Audio Input tab

The audio input settings are the same as the audio output settings except that they apply to audio input. In order to process audio from the audio input in real time a full-duplex sound card (one that can record and playback 16 bit 44k stereo audio simultaneously) is required.

The MIDI & Sync tab

The MIDI & Sync tab allows independent selection of the MIDI devices used for parameter control and MIDI clock synchronisation. The same device may be used for both purposes.

The Directories tab

The paths of wave files referenced within AudioMulch documents are stored in relative form if they are contained within the same directory or sub-directories of the current document, or the AudioMulch application directory.

Additional directories may be specified in the Directories Tab. Directories can be added to the list using the Add... button or removed using the Remove Button.

Note that paths for wave files used in the SoundIn and SoundOut contraptions are always stored in absolute form.

Interim MIDI Documentation

This section provides some instructions for using the currently implemented MIDI features. At present AudioMulch can synchronise to incoming midi clock and song position pointer messages, and can accept midi messages to control most parameter values.

Using MIDI Sync

To use MIDI sync you need to have a sequencer capable of providing MIDI clock and song position pointer information to AudioMulch via a MIDI input device. To begin go to the **MIDI & Sync** tab of the Settings dialog box (**Edit->Settings**) and select the desired **MIDI Sync Device**. Next open a document that contains clock-based contraptions (Bassline, Drums, etc.). Enable external sync by choosing **Chase Sync** from the **Control** menu. Enable real-time audio by clicking the speaker button. Now when you start and stop the external sequencer AudioMulch will start its own clock and synchronise with the external source, it should also track tempo changes.

AudioMulch will take a small time to lock synchronisation, for this reason it is advisable to send a 1 bar count-in to ensure clean sync.

Using MIDI Parameter Control

Most numeric contraption parameters may be controlled from a MIDI control source (sequencer, control surface, etc.). To begin go to the **MIDI & Sync** tab of the Settings dialog box (**Edit->Settings**) and select the desired **MIDI Controllers Device**. Load a document, start the audio and enable midi controllers using the **Enable MIDI Controllers** item in the **Control** menu.

There are two ways to map a MIDI controller to a contraption parameter *Quick Mapping*, and by using the **Parameter Modulation** dialog.

The Parameter Modulation Dialog

The **Parameter Modulation** dialog is accessed by selecting **Parameter Modulation...** from the **Control** menu. On the left of the dialog a tree displays all of the contraptions in the current document. By expanding a node (click on a + or double click a contraption in the tree) all of the controllable parameters for that contraption are displayed. When a parameter is selected in the tree (by clicking on it) modulation information for that parameter is displayed on the right. You can select a MIDI controller type, number and channel. If MIDI controllers are enabled from the Control menu, you can click the button with the midi lead on it and the next controller received will be entered as the control source – this could be used to quickly select a knob on a control surface by simply moving it.

The parameters list displays Icons showing a small midi lead for all parameters that have a MIDI modulation source assigned to them.

You can also display the Parameter Modulation Dialog by right clicking on any knob, slider or other parameter control in a contraption property editor and select **Setup Modulation...** from the popup menu. This will display the Parameter Modulation Dialog with the parameter you just clicked on selected and ready for mapping.

The parameter modulation dialog box is most useful when a number of parameters need to be mapped at once, in the future it will also provide other functionality not available when using quick mapping.

Using Quick Mapping

By right clicking on any knob, slider or other control in a contraption property editor and selecting **Quick-map MIDI Controller...** a simplified version of the Parameter Modulation dialog is displayed allowing a midi control source to be set for the control just clicked. If MIDI Controllers are enabled (from the Control menu) the Quick-map dialog will be waiting for an incoming midi control message – if one is received it will be mapped to the chosen parameter and the Quick-map dialog will close automatically.

Optimising Real-Time Performance

When AudioMulch is installed it is configured to perform real-time signal generation and processing on most systems. Due to various factors, optimal real-time performance can only be achieved by 'tuning' AudioMulch on a specific system. It is worth taking the time to tune AudioMulch to take advantage of the capabilities of your system – the program can be much more responsive than when using the default settings.

This section discusses 'optimal performance' in terms of two parameters: *latency* and *stability*. Stability is concerned with the reliability of the audio input and output – a stable system will deliver hours of continuous sound without a problem, an unstable system will contaminate the audio stream with clicks, pops, stutters and drop outs. Latency manifests as delays between audio entering your soundcard and emerging from your soundcard after having been processed by AudioMulch (audio latency), and as delays between modifying parameter values on screen or via MIDI and hearing the results (control latency). The lower the latency, the more useful the system is for real-time performance, as the system will be perceived to be operating 'in the moment' rather than with a perceptible delay. Achieving a stable system with the lowest possible latency should be your goal when tuning AudioMulch.

The main cause of stability problems is external programs that perform periodic disk activity. The following are commonly known to interfere with the stable operation of digital audio software.

- System Agent (included in the Microsoft "Plus!" pack) can be set to schedule disk scanning and defragmentation at any time – avoid having it wake up while using AudioMulch.
- Find Fast for Office 95 / 97
- Screen Savers
- CD-ROM "Auto-Insert" Notification
- Other programs performing processor-intensive operations while using AudioMulch.

Even disk access within AudioMulch such as recording to a file using SoundOut can interfere with stable operation, as both the soundcard and hard disk controller must share the same buss for communication with the computer. As a result it is often necessary to increase buffer sizes to ensure stable operation when disk access is involved.

Once known sources of instability have been reduced to a minimum buffer sizes can be reduced until the audio begins to break up and then eased off until reliable operation is achieved. The lowest in-to-out latency achievable with AudioMulch is about 40ms (using two 2048 byte buffers for input and output.) The speed of the computer, soundcard and quality of soundcard drivers all effect the minimum workable buffer sizes. On some computers attempting to use too few, or too small buffers may cause a system crash.

It is advisable to use the same buffer sizes for input and output. If audio output is generally stable but glitches occur when using audio input it may be necessary to increase the number of audio input buffers.

In general using more buffers produces greater stability. A few larger buffers are usually preferable to a large number of small buffers. However, there are no hard and fast rules that can be applied to all systems. The best thing to do is to experiment.

Ways to get help while you work

While using AudioMulch you can view this help file at any time by selecting **Help Topics** from the **Help** menu or pressing the F1 key.

Help for individual contraptions may be accessed by clicking on the help button (?) in the title bar of the contraption that you are interested in, or by right clicking on a contraption in the patcher and selecting **Help** from the popup menu.

AudioMulch on the Web

The AudioMulch web site is located at:

<http://www.audiomulch.com>

You can reach this site by entering the above URL into your web browser, or by selecting AudioMulch Web Site from the help menu within the AudioMulch program.

The AudioMulch web site always contains up-to-date news regarding the latest AudioMulch version, and a variety of other useful resources.

Sending Suggestions and Bug Reports

If you encounter a problem with AudioMulch, whether it be a bug, or a feature you would like to see included. Don't hesitate to contact the author Ross Bencina via email: rossb@kagi.com.

If you are reporting a bug, please include the following information with your message:

The version of AudioMulch you are using, for example "0.8b2". This information is displayed in the about box, which can be viewed using the Help->About AudioMulch menu item.

A description of the bug, including:

- What you were doing when it happened
- whether you can repeat the bug
- whether the sound in / sound out was active when the bug occurred.

A quick spec of your machine including:

- OS version (win95, win98, NT4.0 etc.)
- CPU and Speed (eg. Pentium II 266)
- Amount of physical RAM (eg. 64MB)
- Soundcard brand and model

Contacting the Author

Ross Bencina can be reached by email at rossb@kagi.com

Or alternately you can send snail mail to:

Ross Bencina
10-12 Warrandyte Rd
Ringwood, 3134
Victoria
AUSTRALIA

Copyright/License/Warranty

Copyright

The AudioMulch software, documentation and web site is Copyright ©1997-1999 Ross Bencina.

VST support based on the Steinberg VST (Virtual Studio Technology) PlugIn SDK. VST and the Steinberg VST PlugIn SDK are Copyright ©1996, Steinberg Soft und Hardware GmbH.

Shareware Registration

AudioMulch is not free software. Subject to the terms below, you are hereby licensed to use this software for evaluation purposes without charge. If you intend to use this software on an ongoing basis, a registration fee of US\$50 (AU\$50 to residents of Australia) is requested. You can register using the Register AudioMulch... menu item within the program, or on the internet via the AudioMulch web site. When payment is received you will be provided with a registration code.

Alpha and Beta Test Versions

Alpha and Beta versions of AudioMulch are distributed for evaluation and testing purposes only. They are configured to expire after a fixed period from their initial release date (30 days for alpha versions, 90 days for beta versions). A valid AudioMulch registration code will prevent expiry. Release versions do not expire.

Distribution

Different distribution policies are maintained for Beta Test and Release versions of AudioMulch. The policies are described below. Beta Test versions can be identified by the presence of a 'b' in their version number such as 0.8b3. The version number is displayed in the About AudioMulch window available from the Help menu within the program.

Release Versions

You are hereby licensed to make as many copies of the Shareware release version of this software and documentation as you wish; give exact copies of the original Shareware release version to anyone; and distribute the Shareware release version of the software and documentation in its unmodified form via electronic means. There is no charge for any of the above.

You are specifically prohibited from charging, or requesting donations, for any such copies, however made; and from distributing the software and/or documentation with other products (commercial or otherwise) without prior written permission.

Alpha and Beta Test Versions

You are prohibited from distributing Alpha and Beta versions of AudioMulch without prior written permission. Alpha and Beta versions should always be obtained directly from the

AudioMulch web site at: <http://www.audiomulch.com/>

Disclaimer of Warranty

THIS SOFTWARE AND THE ACCOMPANYING FILES ARE PROVIDED "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OR MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED.

ANY LIABILITY OF THE PROVIDER WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF PURCHASE PRICE.

History of AudioMulch Changes

10 / 9 / 99

version 0.8b3

- Added four new contraptions: RissetTones, RissetFilters, Nebuliser and Arpeggiator.
- Moved beta contraptions from 0.8b2 into their appropriate categories (Shaper and DigiGrunge to Effects, *Gain and Invert to Mixers.)
- Added MIDI parameter modulation support to a number previously unsupported parameters (Matrix4x4, Matrix8x8 matrix switches, LoopPlayer checkboxes, Bassline waveform and mute, Drums mute and channel enables, RingAM switch between RM and AM, TestGen sine/noise switch.)
- Removed length limitation for loadable soundfiles in LoopPlayer and Drums, now any length file can be loaded. Be careful not to load huge files if you haven't got a lot of RAM.
- Revised the popup menu structure in the patcher. Now different popups appear whether you click on a contraption or empty space. The functionality hasn't changed significantly except that you can now open multiple contraption editors at once by shift selecting them and selecting Edit from the Patcher popup menu. The Delete menu item works now too.
- When a contraption editor is opened it is tiled in the properties pane, rather than the previous method of having them all open on top of each other.
- Moved level meters configuration to a dialog box. This can be accessed by right clicking on any of the level meters, or from the View -> Toolbars -> Configure Level Meters... menu item.
- Increased parameter ranges in *Gain and *ParaEQ contraptions.
- Added a new open soundfile dialog. When selecting soundfiles for LoopPlayer, Drums, SoundIn etc, there is an option to preview the sounds before loading them. Information about the selected soundfile is also displayed.
- Added support for 8 bit soundfiles and soundfiles of any sample rate for all contraptions that use soundfiles except SoundOut.
- Added the ability to specify a sync offset in milliseconds for MIDI clock sync. This allows compensation for delays in the soundcard or MIDI drivers. It also allows for compensation for latency when processing MIDI synced audio through AudioMulch.
- Added "Mapping" tab to the Parameter Modulation dialog which includes: Upper and lower limits allowing MIDI controllers to be mapped on to a partial range of a parameter; A smoothing (slew limiting) parameter to limit the speed at which parameters are changed; and a non-linear mapping curve. Click on the mapping curve to add new points, drag points outside the curve editing pane to delete them.
- Fixed a bug that would cause a crash whenever a Contraption with MIDI modulated parameters was deleted, or a document containing MIDI modulated parameters was unloaded and then more MIDI input arrived.
- Fixed a bug that would cause audio to fail when a document containing feedback was loaded.
- Fixed a number of bugs associated with MIDI clock sync, it should be more reliable now.
- Fixed a bug that sometimes caused an error when using 'Export as Wave File...'
- Fixed a number of bugs in BubbleBlower and LoopPlayer that resulted in intermittent crashes, sometimes when opening a second file, sometimes files weren't closed properly and became inaccessible to other programs.
- Fixed ties bug in Bassline. Now a pattern with all ties will play forever instead of cutting out.
- Fixed 8x8Matrix, now it works.
- Fixed a bug that caused a crash when using the 'Set Value...' popup item on some of the

knobs in the *ParaEQ contraptions.

- Contraptions with no properties such as Busses are no longer listed in the Parameter Modulation dialog as having a Preset Number property.
- It is no longer possible to accidentally cut and paste a second SoundOut contraption.
- Fixed problems with restoring soundfile name parameters in presets with some contraptions (notably SoundIn). You can now store file references in presets and they will recall properly.
- Fixed a bug in loop player that caused a small glitch at the end of each loop cycle, even when 'Stretch' was enabled. Loop player can now loop files perfectly, allowing it to be used for drone notes and other 'critical' looping tasks.
- Fixed bug with numeric edit controls where dragging upwards to increment negative numbers was unreliable.

5 / 6 / 99

version 0.8b2

- Added "Set Value..." item to parameter Knob and slider context menus allowing exact values to be typed in.
- Fixed various problems with VST plugins: Plugins with custom 'bitmapped' user interfaces now display correctly, plugin settings are now saved with the document correctly, presets now work with VST plugins.
- A warning is now displayed when loading documents that contain uninstalled plugins.
- Bugs which would cause crashes when cutting and pasting contraptions, deleting contraptions, opening and closing documents, and quitting have been fixed.
- Bugs related to the SoundIn contraption that would cause crashes or freezes when creating or deleting SoundIn, stopping audio while playing a file with SoundIn, and switching between real-time audio input and use input file have been fixed.
- Fixed bug where the number of output buffers setting was always reset to 4 when AudioMulch was restarted.
- Added "Enable MIDI Controllers" button to the toolbar.
- Fixed problems with configuring MIDI mappings in the Parameter Modulation window.
- MIDI mappings are now saved correctly with the document.
- The Parameter Modulation dialog box now uses icons to indicate which Contraption parameters are modulated by MIDI.
- Range type parameters (e.g. Flanger frequency range) can now be correctly modulated from MIDI control sources.
- The DLGranulator Freeze parameter was disabled in the previous version. It is now enabled and may be switched on and off under MIDI control.
- Fixed a bug that caused the Matrix contraption to use more CPU cycles than in previous versions.
- Fixed bug where documents opened from Windows Explorer didn't display correctly.
- Fixed bug where documents opened from Windows Explorer display with DOS file names in reload menu.

7 / 3 / 99

version 0.8b1

- New and expanded help files
- New Office 97 style dockable toolbars
- Unified "enable audio" instead of separate audio in / audio out buttons and menu items
- Support for up to 16 channels of real-time audio input and output

- MIDI clock and song position pointer synchronisation
- MIDI control of most contraption parameters
- Support for VST plugins
- Cut, copy and paste in the patcher
- ToolTips displaying the current value of many knobs and sliders
- Multiple patch cords can be connected to a single inlet
- Moved Matrix contraptions to Mixers category
- Moved BubbleBlower contraption to Signal Generators category

4 / 12 / 98
version 0.7b6

- Drums and LoopPlayer now support soundfiles at sampling rates other than 44100 hz
- Cells in the Drums and Bassline sequence editor can now be selected using the keyboard arrow keys and toggled on and off using the space bar
- Added code to prevent the user interface freezing when the cpu load approaches 100%. This is enabled by unchecking the "overdrive" checkbox in the Settings->Audio Output dialog box.
- Fixed bug that caused recalled presets to only update the user interface and not the signal processing
- Added transpose button to Bassline
- Added BubbleBlower and Matrix contraptions

11 / 10 / 1998
version 0.7b5

- Added phaser effect
- Altered behavior of range-sliders to allow thumbs to be moved independently without keyboard modifiers
- Added cut/copy/paste to contraption presets
- Cosmetic changes to contraption editor window
- Added new numeric edit control
- Overhauled Export as Wave File functionality to allow exporting arbitrary lengthed segments
- Now supports non integer tempi

6 / 09 / 1998
version 0.7b4

- Renamed I/O menu to "Control"
- Added "Volume Control" to the View menu, this displays the windows volume control
- Fixed 5Combs, now displays correct frequency and note numbers when its property editor is displayed
- Re-coded real-time audio i/o code again to fix glitching problems on slower machines
- Added second input to SDelay
- Fixed bug in Drums which was causing clicks on re-attacked samples

20 / 07 / 1998
version 0.7b3

- Added audio input to Bassline to allow processing through Bassline's filter sequencer
- Added master gain control to Additive10 and renamed it to 10Harmonics
- Cleaned up Bassline inc. accent and sawtooth
- Added an external modulator input to RingAM
- Support for tempos down to 1 bpm
- Fixed NT Audio i/o bugs
- Re-coded real-time audio i/o code, now supports lower latencies and glitch free audio input
- Adjusted mixer code so that moving the knobs doesn't generate zipper noise
- Placed a limiter in the feedback loop of DLGranulator to prevent feedback blow-out problems
- Implemented automatic feedback loop limiting, this prevents patches with feedback loops from crashing with exception 10H, it also sounds a lot better!
- Fixed Find Tab in help file
- Fixed Spat crashing bug
- Fixed other miscellaneous nasty bits
- Added Future.txt file to the distribution

23 / 04 / 1998
version 0.7b2

- Audio input settings can now be updated in real-time
- Basslines no longer intermittently crash program
- "Mulching Contraptions Reference" section of the help file completed
- Improved sound i/o logic that was causing "MMSystem004 device already in use" errors
- Cleaned up (hopefully all) code that was causing "Invalid Floating Point Operation" errors
- Made miscellaneous cosmetic interface changes
- Improved filter code in ParaEQ, Flanger, SDelay etc.
- Added SSpat stereo spatialiser contraption
- "Save pattern to wave file" can now overwrite existing files
- WaveOut device volume is only reset to full if it was zero (previously it was reset every time).
- Now keeps track of current directory independently for document, sound source and sound destination directories
- Removed " "" is not a valid floating point value" error when entering values into numeric edit boxes

24 / 03 / 1998
version 0.7b1

- First public release

Acknowledgements

Without the encouragement and assistance of a number of individuals AudioMulch would never have seen the light of day. I would like to thank the following people for their contribution to this project:

Steve Adam and Michael Hewes for their suggestions and encouragement during the initial phases of development, way back when AudioMulch was "Ross Tools" on the Macintosh.

Dean Walliss for endless object modeling discussions and most of the original Help file.

Andrew Bencina for authoring the current help file.

Neil Davies at 3Deep and Matthew Proctor at Imagineering for managing our web presence.

Gordon Monroe for deriving the constant grain density algorithm used in the BubbleBlower.

Thanks to the kind folks on the music-dsp mailing list for providing an open forum for music software development.

I would also like to acknowledge all of the courageous beta testers, whose submissions have made AudioMulch a more stable program: Adrian Bond, Akin Fernandez, Alberto di Bene, Angelo Farina, Antonio Russek, Arild Buanes, Benjamin Tucker, Brad Derrick, Brian Edwards, Brian Mulvey, Carl Gibbons, Carlos Fesser, Chris DeLaurenti, Chromozone, Dan Fain, Dana Woodaman, Daniel Gendreau, David Arthur, Dean Walliss, Ernesto Cedillo, Fraser Caldwell, Garth Wood, Gary Walker, Gareth Whittock, George Hagegeorge, Graeme Gerrard, Hakan "Zap" Andersson, Jake Almond, Jake Boswell, James Buecler, James Taste, Jesse Gay, Jerome Miller, Jimmy Lewis, Jiminy Lummo, Joel Stern, Jonathan Klizas, José Antonio Alvarez, Karl Foster, Kelly Allen, Matt Tracy, Matt Wand, Martin Fidibus, Mat Morgan, Michael John Trettewick, Michael Moncur, Mike Chapman, Mirjan Milovanovic, Mr Thinly Sliced, Nicolas Fournel, nic linfernel, nrshogia, Oliver Sparrow, Oshine Najarian, Paul, Peter Batchelor, Phil Burk, Philippe Caya, Paul Fravigar, Peter Remmers, Rainer Linz, RexBo, Reinhard Hoerl, Richard Burden, Robert Maroney, R. Wolf, S., Sam Jorgensen, Semtex, Shawn Hatfield, Stephen Graham, Steve Chalberg, Steve Hamann, Steven Hugg, Stefan Reich, Tomas Jirku, Toni Van Remortel, tyler, Warren Burt, Gareth Whittock, Tim Robinson, Beatrice Jasper, Yasiu.

If I've missed anyone you know who you are – thank you.

About the Author

Ross Bencina is an electroacoustic composer, and a programmer. Ross' works have been performed in the southern states of Australia and broadcast nationally. His music incorporates live improvisation with acoustic instruments combined with real-time audio signal processing and pre-recorded sound composition. He has published papers on computer music software development and human/computer performance aesthetics. Ross currently resides in Adelaide, South Australia where he works as a freelance multimedia developer when not composing or hacking AudioMulch.

Selected Technical Bibliography

Below is a list of the books and articles that have been useful while writing AudioMulch. The list is not complete, notable exceptions include many miscellaneous articles in *Computer Music Journal* (MIT Press), *Proceedings of the International Computer Music Conference (ICMC)* and *Doctor Dobbs Journal*. The music-dsp mailing list has also been an indispensable resource, you can find out about the music-dsp mailing list at their web site: <http://shoko.calarts.edu/~glmrboy/musicdsp/music-dsp.html>

For a more musically oriented reading list, see Further Reading in the Tutorials section.

Computer Music

Moore, F.R. 1990. "Elements of Computer Music," Prentice Hall, Englewood Cliffs, New Jersey.

Orfanidis, S. J. 1996. "Introduction to Signal Processing," Prentice Hall, Upper Saddle River, New Jersey.

Puckette, M. 1988. "The Patcher," Proceedings, ICMC. San Francisco: International Computer Music Association, pp. 420-429.

Roads, C. 1996. "The Computer Music Tutorial," MIT Press, Massachusetts.

Truax, B. 1988. "Real-Time Granular Synthesis with a Digital Signal Processor." *Computer Music Journal* 12(2): 14-26.

General Programming

Booch, G. 1995. "Object-Oriented Analysis and Design, 2nd Ed." Benjamin/Cummings, Redwood City, California.

Martin, R. C. 1995. "Designing Object-Oriented Applications Using the Booch Method," Prentice Hall, Englewood Cliffs, New Jersey.

Sedgewick, R. 1992. "Algorithms in C++," Addison-Wesley, Reading, Massachusetts.

Stroustrup, B. 1991. "The C++ Programming Language, 2nd Ed." Addison-Wesley, Reading, Massachusetts.

Szyperski, C. 1998. "Component Software – Beyond Object-Oriented Programming," Addison-Wesley Longman Ltd., Edinburgh Gate, Essex.

Source Code

Listed below are some freely distributed sources of musical signal processing code. Although I have not used this source code in AudioMulch, I learnt most of what I know about implementing audio signal processing algorithms from studying this code.

Csound software sound compiler

http://www.leeds.ac.uk/music/Man/c_front.html

Princeton CMIX sound processing toolkit

<http://silvertone.princeton.edu/winham/cmix.html>

NeXT / CCRMA Music Kit

<http://www-ccrma.stanford.edu/CCRMA/Software/MusicKit/MusicKit.html>

