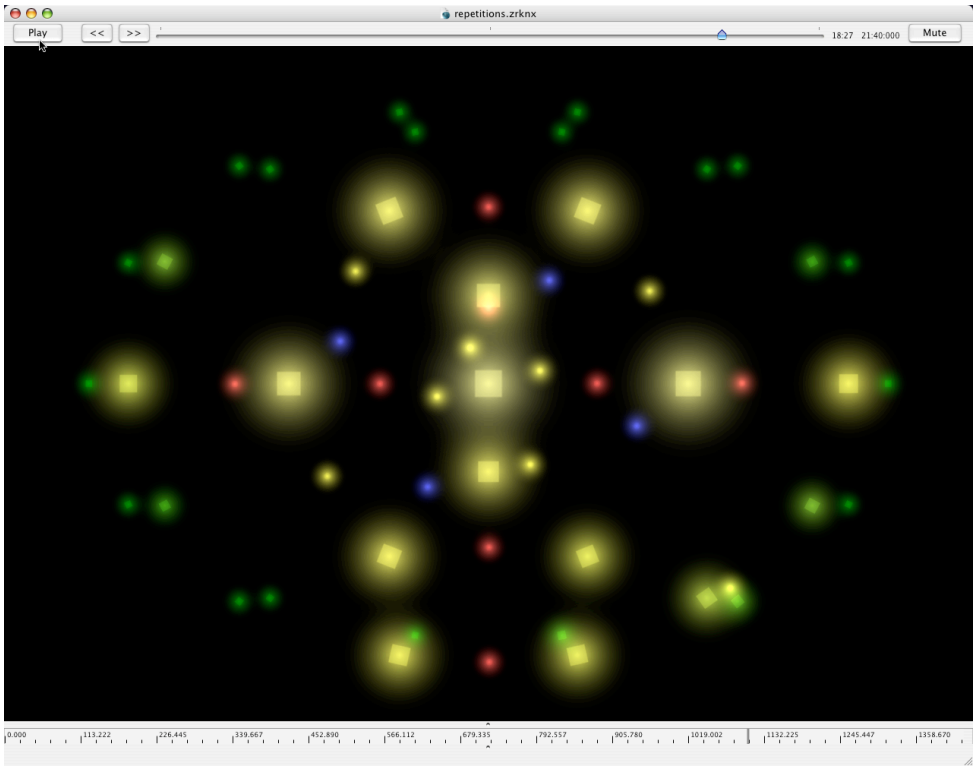
ZIRKONIUM



ZKM | Institut für Musik und Akustik

Chandrasekhar Ramakrishnan
cramakri@zkm.de

TABLE OF CONTENTS

ZIRKONIUM

ZKM | Institut für Musik und Akustik

Chandrasekhar Ramakrishnan

cramakri@zkm.de

March 2007

Overview

Klangdom in the Kubus

Zirkonium is a program for sound spatialization. It was originally developed for the ZKM Klangdom, a 47 loudspeaker environment installed in the ZKM's Kubus, but it has designed to be used in conjunction with out loudspeaker environments as well.

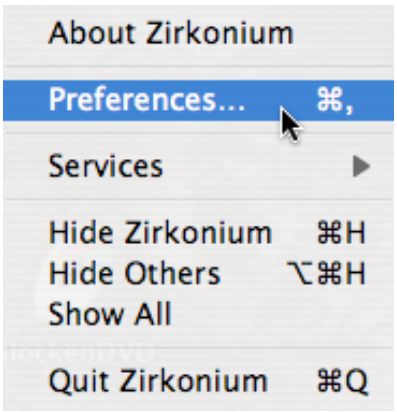
The aim of the software is to simplify the use of space as a compositional parameter. Zirkonium can be used as a stand-alone application for creating a multi-track , spatialized composition or simply as a tool for spatialization within another program. The capability to read and playback sound files in a variety of formats (including AIFF/AIFC, WAV, SDII, SND, MP3, AAC) with arbitrary numbers of channels is built in, as is the ability to incorporate live input from an audio device. Zirkonium also provides the means to compose, store, and playback a spatial choreography. In cases where there is no need for realtime audio processing or algorithmic control of sound source position, the entire piece can be realized within Zirkonium itself. For more complex situtations, Zirkonium can present itself as an audio device to other programs and can accept control data via Open Sound Control (OSC). This lets the user work in their preferred environment (Digital Performer, Logic, Max/MSP, SuperCollider, etc.), using Zirkonium for multi-channel panning.

Configuration / Preferences

Zirkonium differentiates between two kinds of information in the form of different document/file types. One type of document is the description of a piece, which has the extension *zrkpxml*. These are created by the user to define a spatialized musical composition.

The other type of document is the studio configuration document, which has the extension *zkmstu*. This document is created by the software itself and is used to define the environment the software runs in. The file is stored in *~/Library/Application Support/Zirkonium/studio.zrkstu* and contains contains information pertaining to the loudspeaker configuration, preferred audio interface, etc. — information that is dependent on the setup in a particular room.

To edit the studio setup, invoke the Zirkonium preferences menu item.



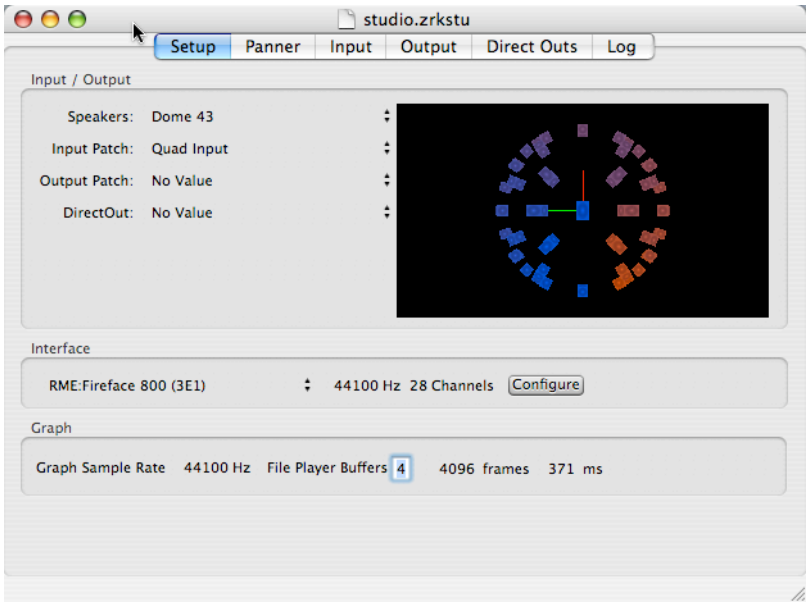
The preferences panel has several tabs, *Setup*, *Panner*, *Input*, *Output*, *Direct Outs*, *Log*, each focusing on a particular aspect of the overall configuration. The first tab, Setup, presents the parameters that define the behavior of Zirkonium. The other tabs provide interfaces to create input/output/direct out patches (more on this below) and test the panner.

Setup

The parameters on the setup tab configure the behavior of Zirkonium.

Speakers

The speaker setup specifies the number and location of the speakers. Zirkonium is delivered with several predefined speaker configurations: quadraphonic, 5.0, octophonic, Dome 4, Dome 37, and Dome 43. Using the Studio->Studio Setup menu, it is possible to add alterna-



Setup Preferences

tive speaker configurations (note, this functionality is not available in all versions).

Input Patch

Input patches and output patches provide a way of reordering the input and output channels from the audio device. If no input patch is specified, all the channels from the device are made available to Zirkonium in the order in which the device presents them. E.g., Zirkonium input channel 4 is device input channel 4. This is not always ideal. Sometimes, only a subset of the channels are needed, but with a different ordering. Using an input patch, Zirkonium can be configured to make device channels 1, 7, 8, 9 available as channels 1, 2, 3, 4. The details are explained in the section on input patches below.

Output Patch

The output patch is similar. It defines a mapping from channels in Zirkonium to channels on the device. This is necessary when the channel ordering internal to the software does not the one on the device. The section on channel numbering contains additional information on this topic.

Direct Output Patch

A direct output patch lets the user bypass the panner and route channels directly to outputs on an audio device. By default there are no direct outs, but selecting a direct out patch turns this capability on and defines the routing of direct out channels to device outputs.

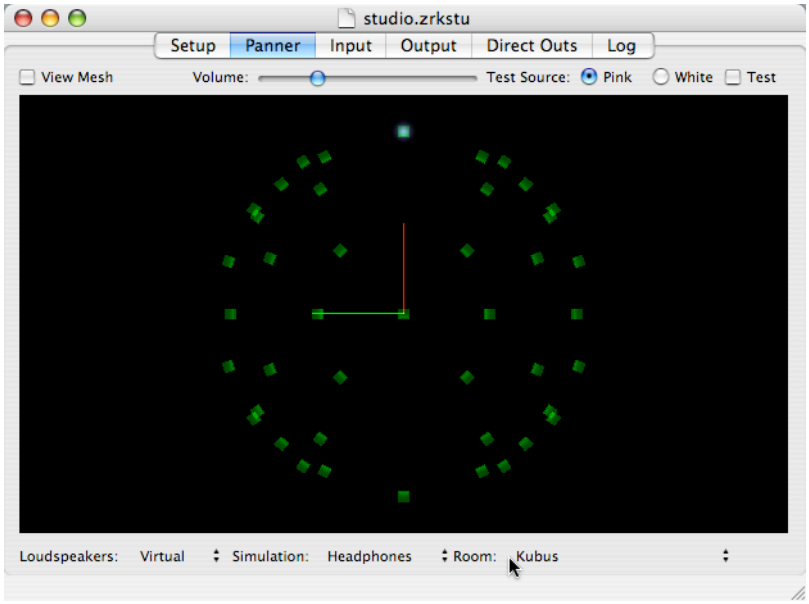
Interface

Selects an audio interface for Zirkonium to use. The *Configure* button launches the designated application for editing the devices settings (sample rate, etc.).

Graph

Shows information about the internal graph that Zirkonium uses. If glitches are experienced during playback of audio files, increasing the number of internal buffers may help. The overall size of the cached audio file data in milliseconds is displayed based on the number and size of the individual buffers. The default number of buffers is 4.

PANNER



Panner Preferences

Zirkonum uses the Vector Base Amplitude Panning (VBAP) algorithm described by V. Pulkki (Pulkki 1997) to implement the spatialization. VBAP is an extension of the well-known equal-power panning for stereo, but instead of using two loudspeakers, VBAP pans a sound source between the three nearest loudspeakers (in general) to create the illusion of a sound source where no speaker actually exists. Zirkonium additionally provides the option of simulating a loudspeaker configuration for experimenting with loudspeaker setups requiring more loudspeakers than are available.

VIEW MESH

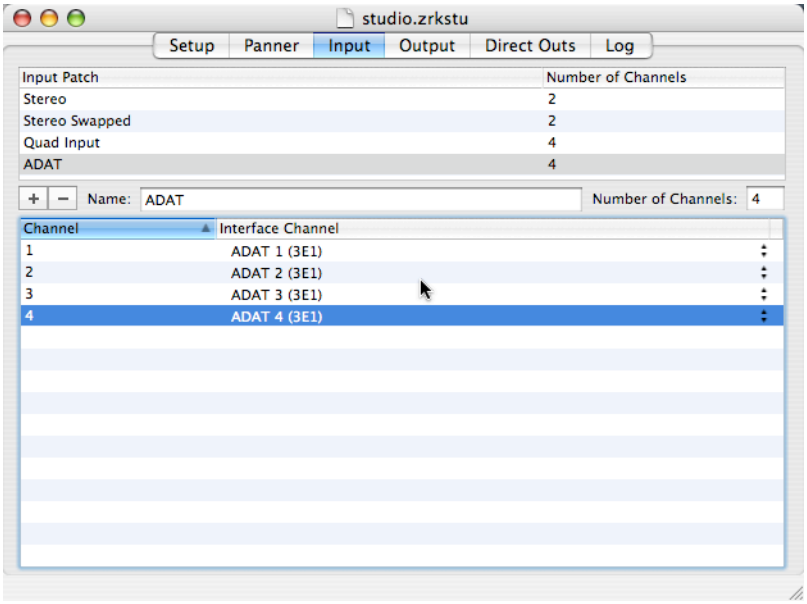
VBAP pans a sound source between the three closest speakers to create the illusion of a sound source where there are no speakers. The *View Mesh* option displays the groups of three loudspeakers using by the panning algorithm.

TEST MODE

The test mode creates a virtual sound source of either pink or white noise that can be panned around with the mouse. This can be used to try out a particular speaker configuration to make sure it is working correctly. It can also be used to experiment with a loudspeaker simulation to develop a feel for how the simulation sounds. The *Volume* slider controls the volume of the test source.

LOUDSPEAKER MODE

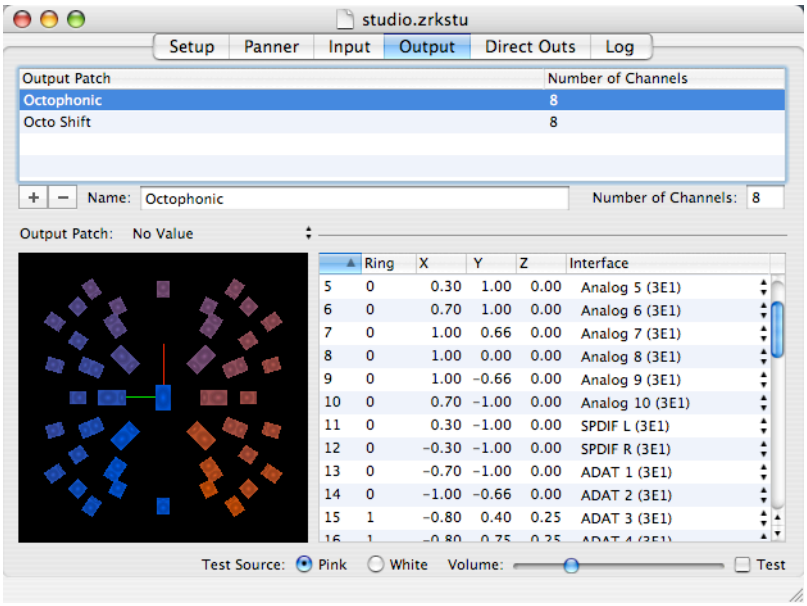
The *Real* mode configures the panner to pan for actual loudspeakers; that is, the device has enough outputs to accommodate the number of speakers in the configuration and the speakers are where they are specified in the configuration. The *Virtual* mode activates a simulation, which can be configured either for headphone listening (using HRTFs for the simulation) or for 5.0 listening. The room option determines the distance of the virtual loudspeakers from the virtual listener.



Input Preferences

INPUT

The input patch determines the routing from a device's inputs to Zirkonium inputs. By default, all inputs from the device are available to Zirkonium pieces. If only a subset of the inputs are actually used, the unused inputs can be turned off to reduce processor load. In the example image, the user has set up an ADAT input patch which makes ADAT channels 1-4 available as the first four channels in Zirkonium. This saves on data throughput by only using the 4 channels that are actually required.



Output Preferences

OUTPUT

The output patch plays a similar role to the input patch, but is used primarily for re-routing, not to reduce processor utilization. Zirkonium uses an internal numbering of loudspeakers that may not match the numbering scheme used in the studio. In that case, the internal numbering, presented in the first column of the table, can be re-routed to the device's output connected to that speaker.

OUTPUT PATCH

The table shows the routing of the currently selected output patch. The process of creating an output patch is unfortunately a bit confusing, but it works as follows. First add a new patch

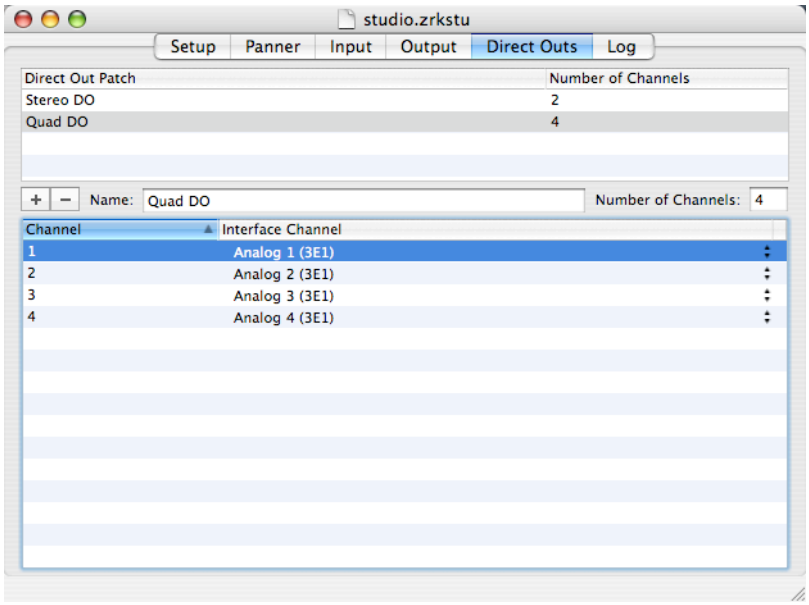
by using the plus button and set the number of channels it uses. Then select it in the *Output Patch* drop down — this activates it in the table. Clicking on a loudspeaker in the dome view will highlight that speaker in the table, simplifying the process of locating the speaker. It can then be bound with the desired output on the device.

Test Mode

Similar to the panner's configuration, the output patch can be switched to test mode. In this mode, the selected loudspeakers receive a continuous noise input (pink or white, depending on the selection) and one can check that the output patch works as desired.

Direct Out

Direct outs provide a way to bypass the panner and route a sound source directly to an output on the device. For example, a composition with click track would use a direct out to route the click track to the preformer's headphones.



Direct Out Preferences

Log

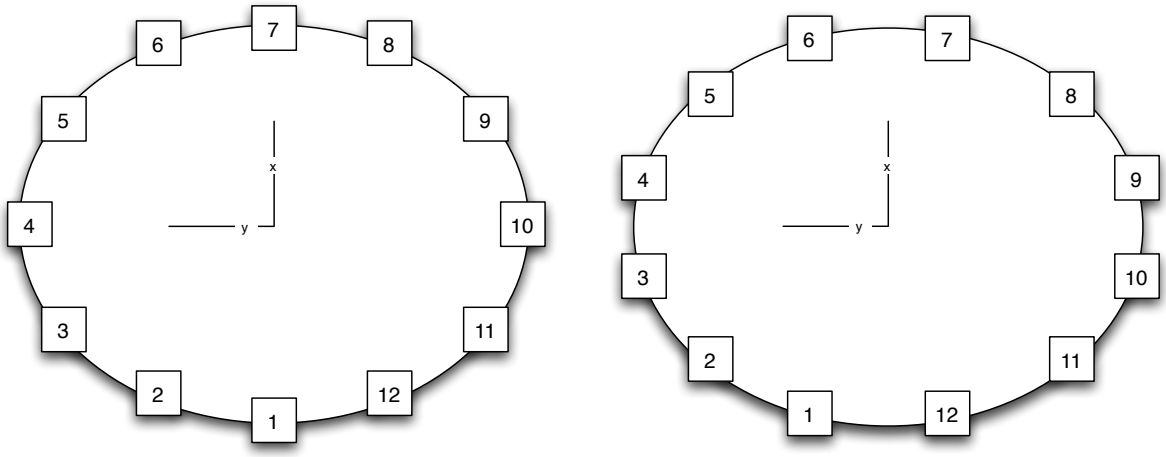
The log is used for internal debugging purposes.

Channel Numbering

In stereo systems, there is a standard numbering of the channels: 1 -> left, 2 -> right. But once we move to quadraphony, the situation becomes more complicated. There are at least two commonly used conventions: left front (LF), right front (RF), left rear (LR), right rear (RR) and LF, RF, RR, LR. Octophony has a few commonly used orderings, but beyond eight channels, all bets are off.

To avoid this confusion, we use a number convention that does not match any of the aforementioned systems, but is consistent and scales to an arbitrary number of speakers:

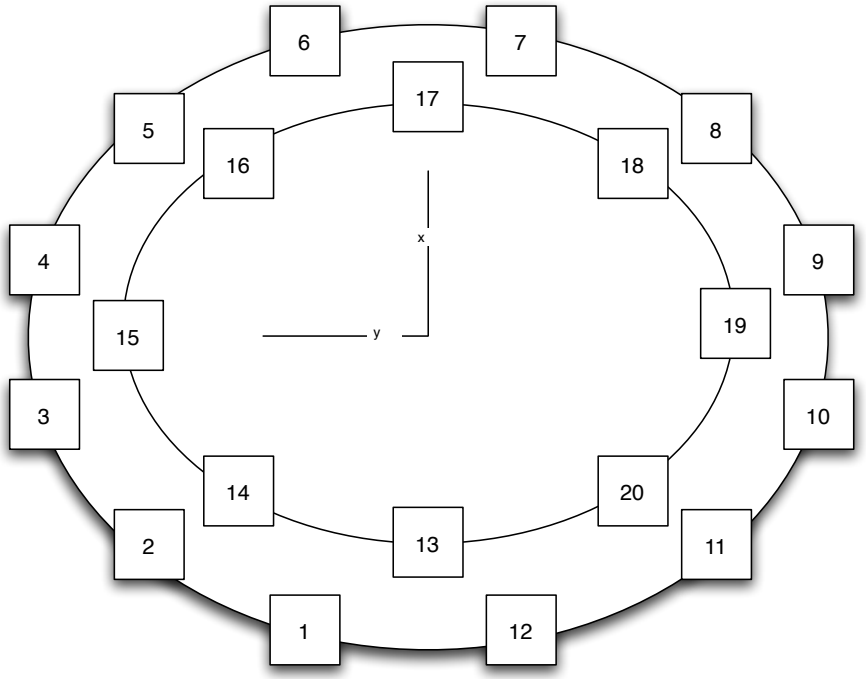
- The speaker directly behind a listener in the center of the setup is numbered speaker 1. If there is no speaker directly behind the listener, then the first speaker behind and to the left.
- Speakers are successively numbered from 1 in the clockwise direction.



Loudspeaker Ordering

If there are multiple rings of loudspeakers, then we can extend the same convention to accommodate that situation. Within each ring, we count the same as above. The first speaker in the bottom-most ring is 1, the first speaker in the second ring is *number of speakers in ring 1* + 1, etc.

The description sounds confusing, but looking at the examples should clear up any



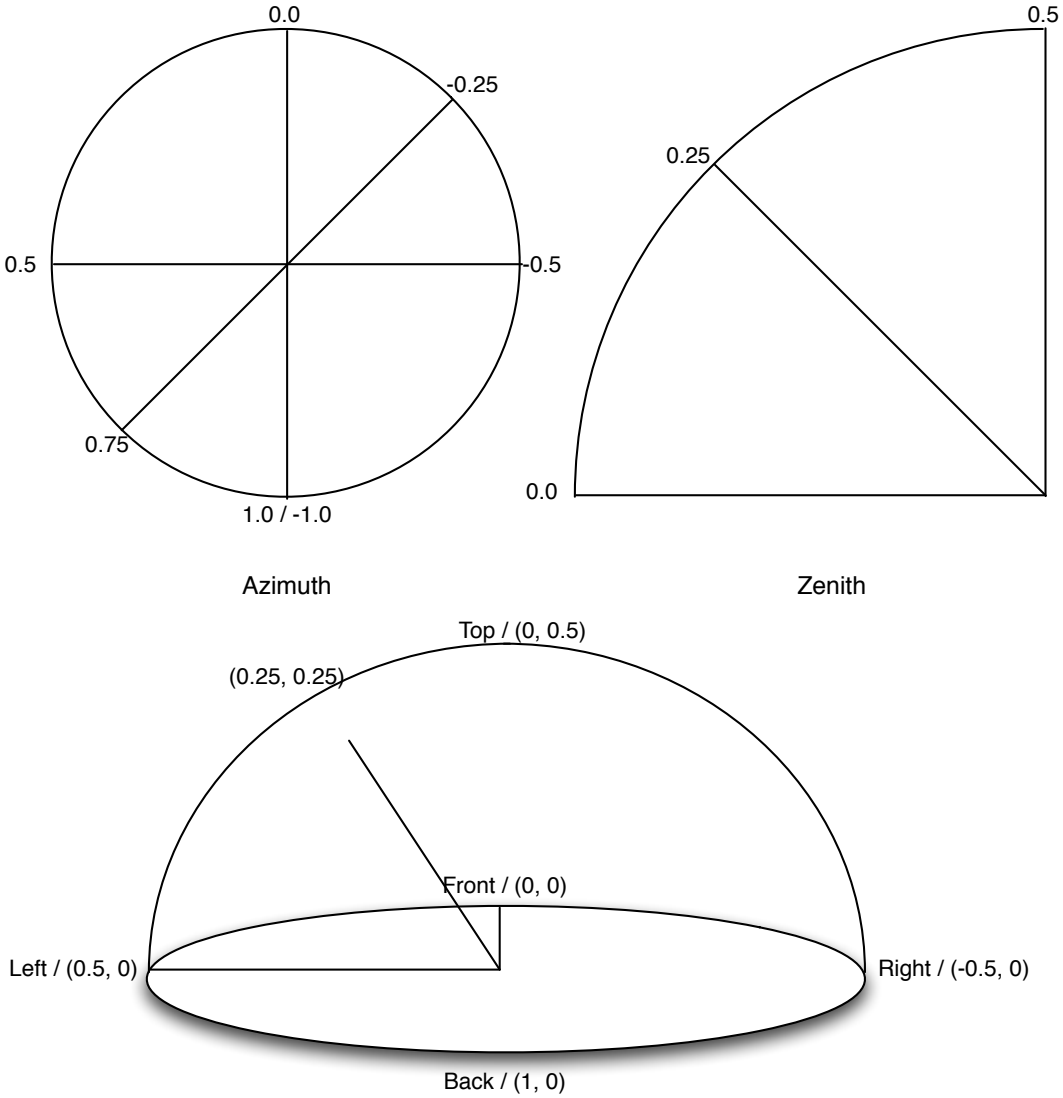
Loudspeaker Ordering with Multiple Rings

issues.

Azimuth and Zenith Units

Zirkonium uses spherical coordinates to specify the position sound sources. VBAP assumes that all speakers and sound sources are on the surface of the unit sphere, thus radius is always 1.0. Zirkonium uses radians/π as the unit for azimuth and zenith since that simplifies conversion to radians and degrees (both of which are used

on occasion internally). To convert to radians, simply multiply by π, and to convert to



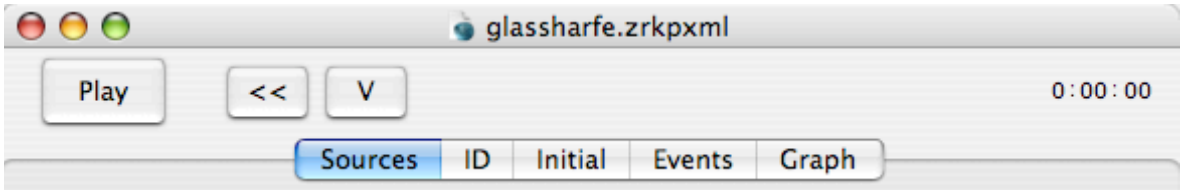
Azimuth / Zenith

degrees, multiply by 180.

Piece

A Zirkonium piece defines a spatialized composition. The piece document is stored in an XML-based text format with the extension *zrkpxml*. In simple cases where the composition is made up of sound files and/or live input with pre-defined movements, Zirkonium is the easiest way to create a spatialization. In cases where live processing of audio is necessary, movements are computed algorithmically, or control data is captured from an external source (such as sensors, a Wacom Tablet, a Lemur, etc.), another program must be used in addition to Zirkonium.

The piece user interface is used to define sound sources for spatialization, a mapping from sound sources (which may be multi-channel) to mono IDs that can be positioned, and it is used to define the spatial movements of those IDs. At the very top of the piece UI, there are several elements that control playback and visualization.



Piece UI

PLAY/STOP

Starts or stops the playback of a piece.

<<

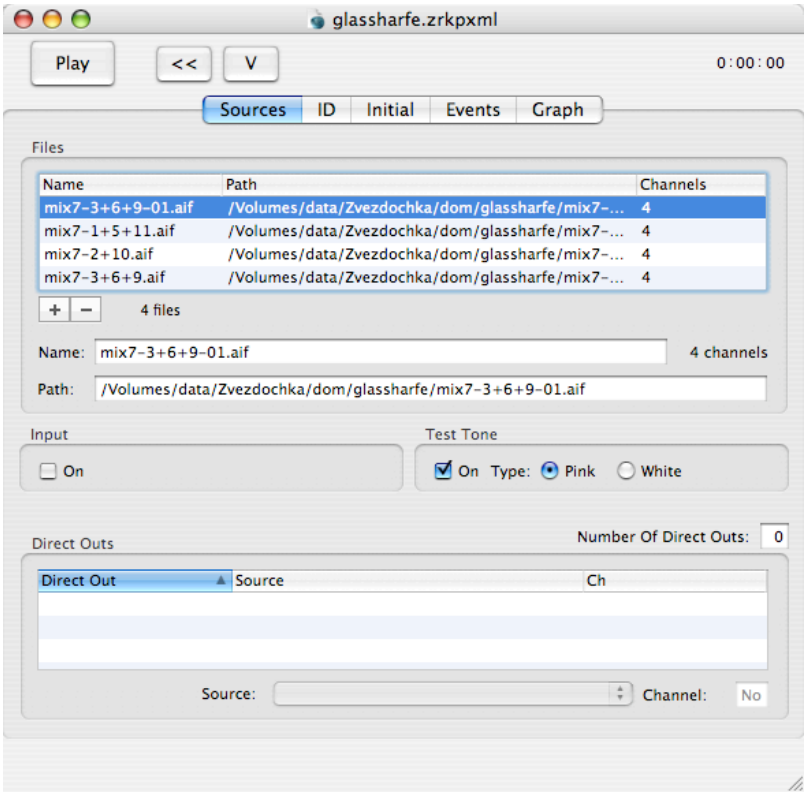
The *rewind* button jumps to the beginning of the piece (i.e., time := 0:00:00).

V

The *V* button opens up a visualizer in a separate window.

TIME DISPLAY

The time display shows the current position in the piece as Hours:Minutes:Seconds. This is editable and changing the value will move the transport to that position. Note, at the moment, this value can only be changed if the piece is stopped; editing is disabled while the piece is playing. This will hopefully be improved in the future.



Piece Sources

Sources

Sources are where the piece retrieves its audio data from. Sources are either sound files or audio input. Test tones are also offered as sources, but this is simply for testing purposes.

Note: the word ID is used to refer to mono sound objects (i.e., mono sound sources) that are spatialized to avoid confusion in the use of the overloaded word „source“.

Files

The files that make up the spatialization. Files can be in any format supported by QuickTime (AIFF/AIFC, WAV, SDII, SND, MP3, AAC, among others) and can contain any number of channels. Files can be added to the piece either via the + button or by drag-and-drop of a file from the Finder.

Input

Turns on the device input. If an input patch is selected in the preferences, it is used to define the numbering of the channels. Otherwise, the channel numbering is the same as on the device.

Test Tone

Turns on a test source as a spatialization sound source. The test tone only has one channel.

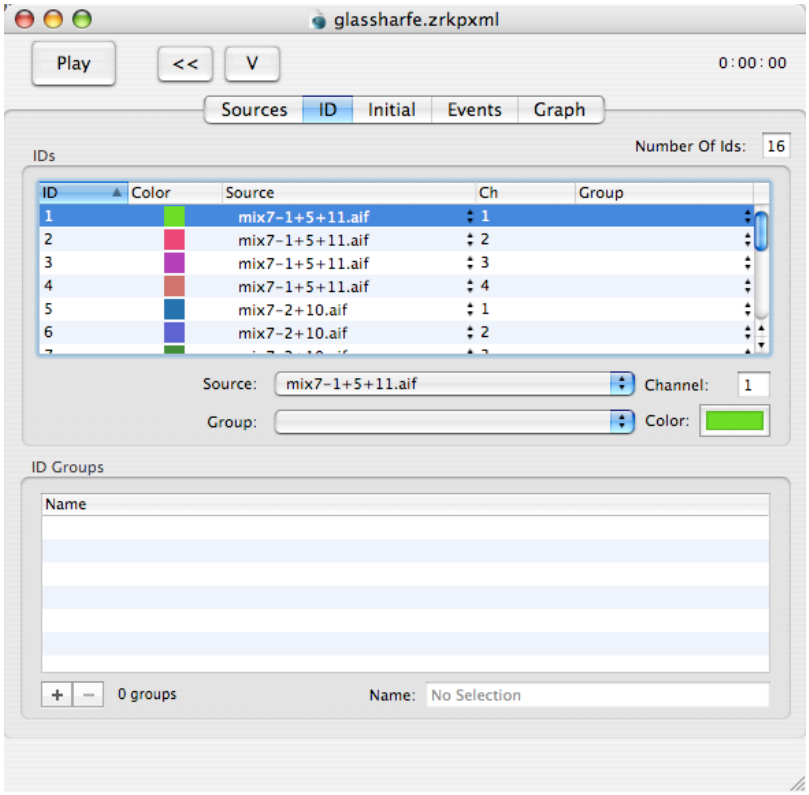
Direct Outs

Binds a sound source to a direct out. The direct outs bypass the panner and are routed directly to one speaker. The routing is dependent on the direct output patch selected in the preferences. Direct Out 1 is mapped to channel 1 of the direct output patch; Direct Out 2 is mapped to channel 2 of the direct output patch, and so on.

ID

IDs are identifiers for mono sound objects that can be panned and are associated with a sound source and channel on that source. The same source/channel combination can be assigned to multiple IDs, if desired.

IDs may also be grouped together to facilitate simultaneous movement of multiple IDs.



Piece ID

Number of IDs

Sets the number of objects to spatialize.

Source

The source (sound file, input, or test tone) the ID gets its data from.

Channel

The individual channel on the, possibly multi-channel, source to narrow down the association to a single mono data source.

Group

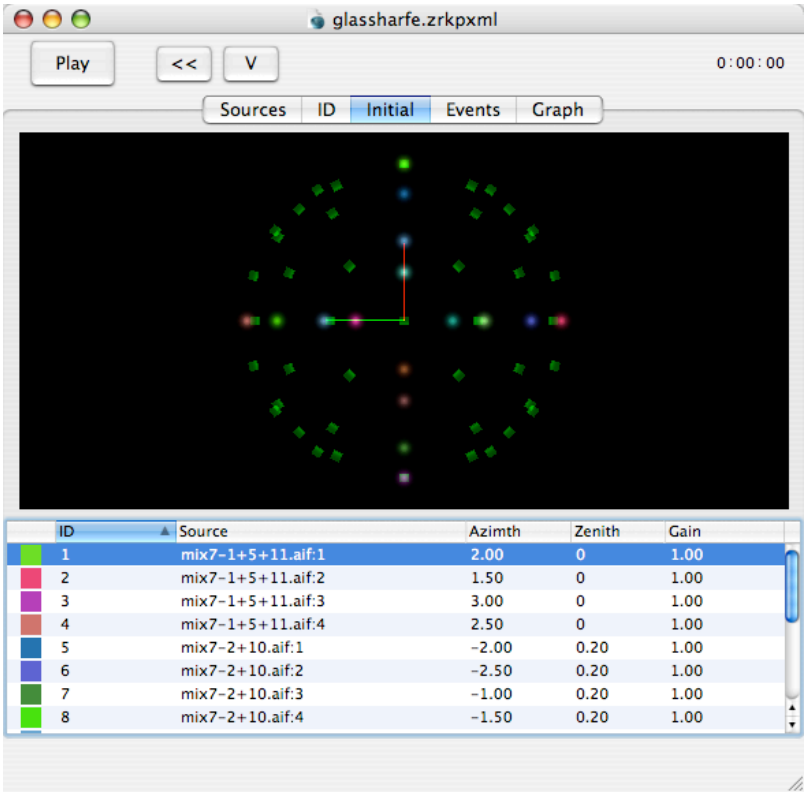
The ID Group this channel is associated with, if there is one.

Color

The color used in the visualization for this ID.

ID Groups

Adds or removes ID Groups.



Piece Initial

INITIAL

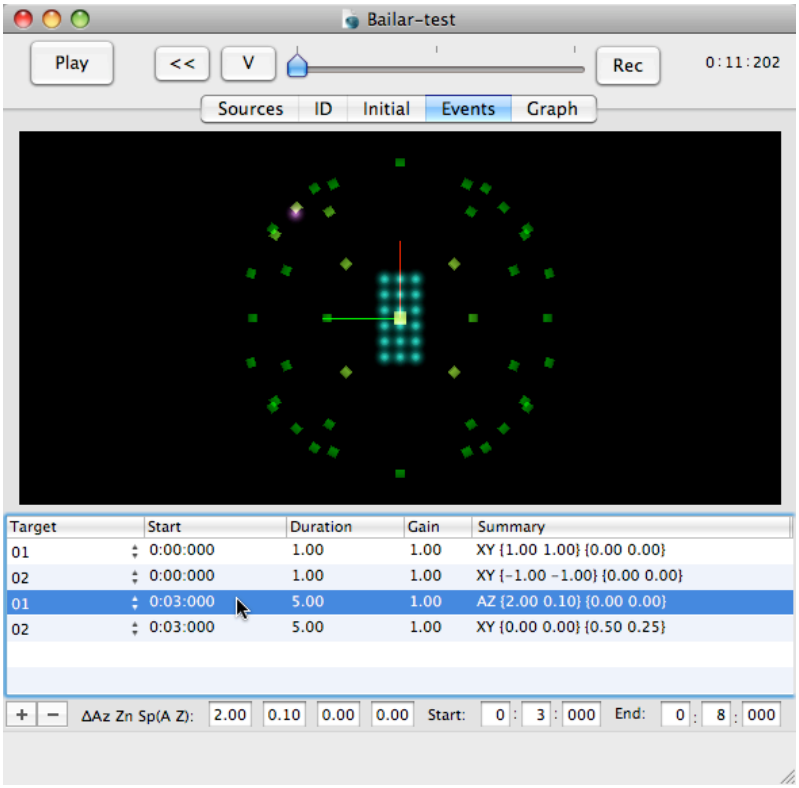
The initial tab defines the starting positions of the IDs. The initial positions are stored separately from the events to make it easy to define a piece in which all the positions are static. For such a composition, the user need only define the initial positions of the IDs.

SPATIALIZER VIEW

Shows the location of the IDs. IDs can be repositioned via the mouse in this view.

ID TABLE

A tabular view of the initial positions and gains of the IDs.



Piece Events

EVENTS

An event defines a movement of an ID. Events are created and deleted via the + an - buttons, respectively. Events may take on one of two types: Spherical or Cartesian. Spherical events use spherical coordinates (azimuth and zenith). Cartesian events use Cartesian coordinates (x and y).

TARGET

The ID or Group the event refers to.

Start Time

When the movement begins. In the table, start time is displayed in seconds; in the text fields, start time is displayed in an SMPTE-like format: MM:SS:MS.

Duration

How long the movement lasts. The duration is displayed in seconds in the table and as an end time in the text fields below.

Summary

Summarizes the details of the event. The format depends on the event type. The summary has one of the following formats: AZ {delta azimuth, delta zenith} {azimuth span, zenith span} or XY {final x, final y} {x span, y span}.

Delta Azimuth

The *change* in azimuth from the beginning to the end of this event specified in radians/π. This information is stored as a delta so it can be applied to ID Groups as well as single IDs.

Delta Zenith

The *change* in zenith from the beginning to the end of this event specified in radians/π.

Azimuth Span

The spread or fuzziness of the azimuth. An ID with an azimuth span of 0.0 is spatialized inside one group of three loudspeakers. By using the span, the sound can be made broader and be diffused from multiple groups of three loudspeakers. Ranges from 0 (a point source) to 2.0 (an entire circle).

Zenith Span

The spread or fuzziness of the zenith. Ranges from 0 (a point source) to 0.5 (a 1/4 circle).

X

Unlike in the spherical case, x is an absolute value, not a delta value. X is the final x coordinate for the event.

Y

Unlike in the spherical case, y is an absolute value, not a delta value. Y is the final y coordinate for the event.

X Span

The spread or fuzziness of the x. Ranges from 0 (a point source) to 2.0 (an entire circle).

Y Span

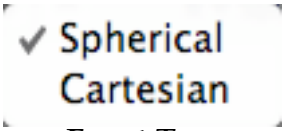
The spread or fuzziness of the y. Ranges from 0 (a point source) to 2.0 (an entire circle).

Gain

The gain applied to this ID. 1.0 leaves the volume unchanged.

Add

The contextual menu (accessed by a right-click or ctrl-click) on the add button makes it possible to select the type of event to add, spherical or Cartesian.



Event Type

Ranges Summary

Parameter	Min	Max	Notes
Azimuth	-∞	+∞	Wraps every 2.0 units.
Zenith	0.0	0.5	
Azimuth Span	0.0	2.0	
Zenith Span	0.0	0.5	
X	-1.0	1.0	
Y	-1.0	1.0	
X Span	0.0	2.0	
Y Span	0.0	2.0	
Gain	0.0	+∞	

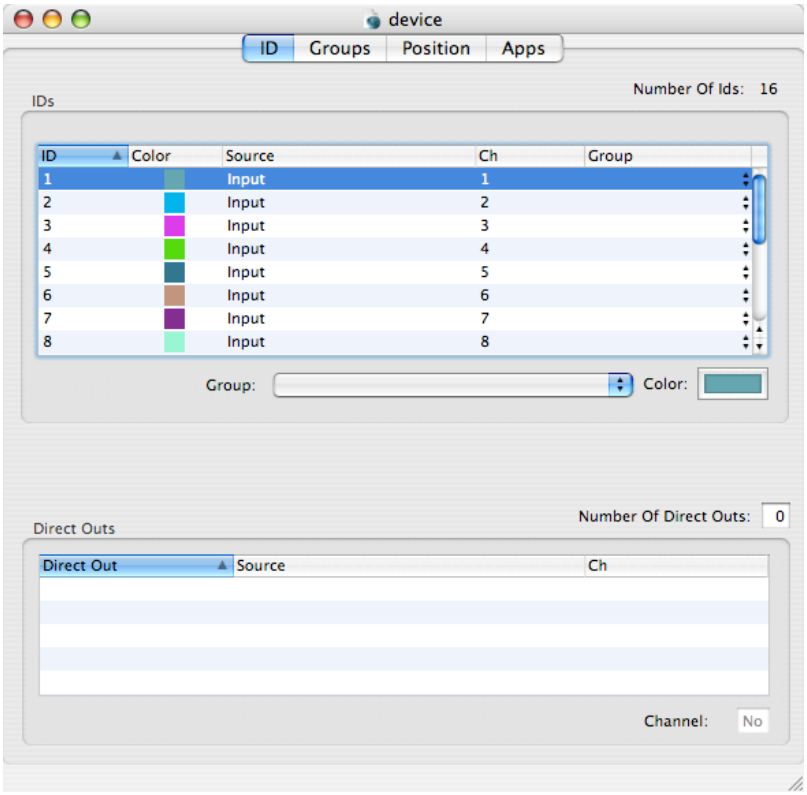
Graph

Used internally for debugging.

Device Mode

One of the design goals of Zirkonium is that it should be a tool, not a platform. That means, Zirkonium should focus on doing one thing well, panning, and not require users to do all their audio generation, processing, or editing inside Zirkonium. Given the current landscape of audio programs, it is unreasonable to expect one application to satisfy all possible needs, thus Zirkonium offers the device mode as a way for other applications to transmit audio to the panner. This allows a user to create a patch in Max/MSP, for example, and use the Zirkonium panner for spatialization.

When Zirkonium is running, it appears to other programs as an audio device. Note: some programs, such as Max/MSP, must be started after Zirkonium because they do not have the capability of adapting to the existence of a new device once running.



Device ID

ID

Just as with pieces, the ID is the unit of spatialization. Zirkonium presents itself as an interface with 16 inputs and outputs. The inputs take audio from the underyling audio device (configured in preferences) and forward that to any client programs. The outputs appear as IDs in the device mode.

IDs

The IDs for the device. The number of ids + the number of direct outs is fixed at 16. Adding a direct out takes away an ID and removing a direct out adds an ID.

Group

The group the ID is associated with. Groups are defined in the group tab.

Color

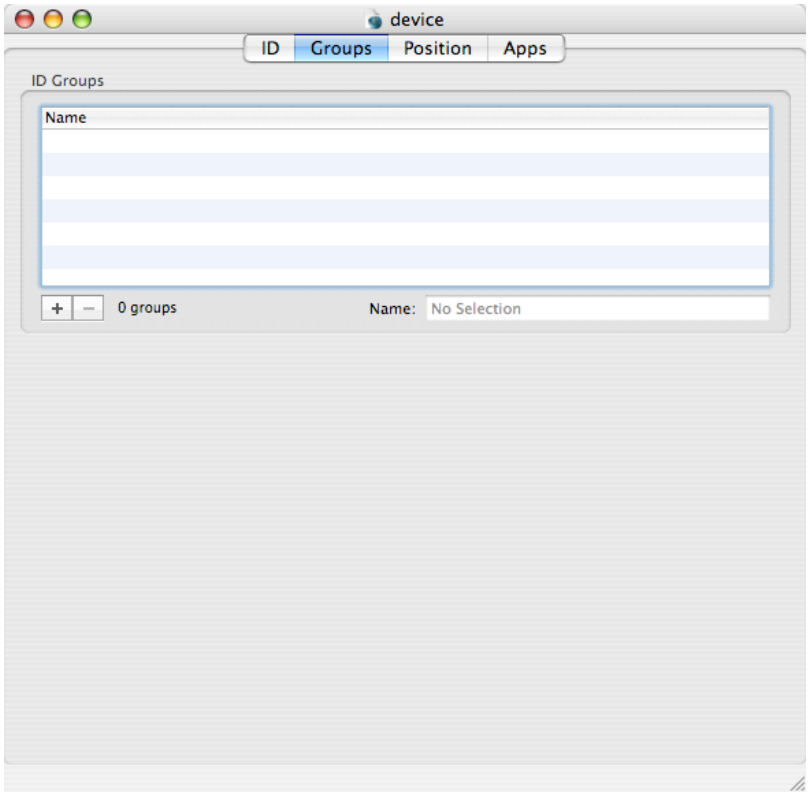
The color used to draw the ID in the position tab's spatializer view.

Direct Outs

Offers a means of bypassing the panner. Since the number of inputs is fixed at 16, adding a direct out removes an ID. The routing for direct outs is determined by the active direct out patch (set in preferences).

GROUPS

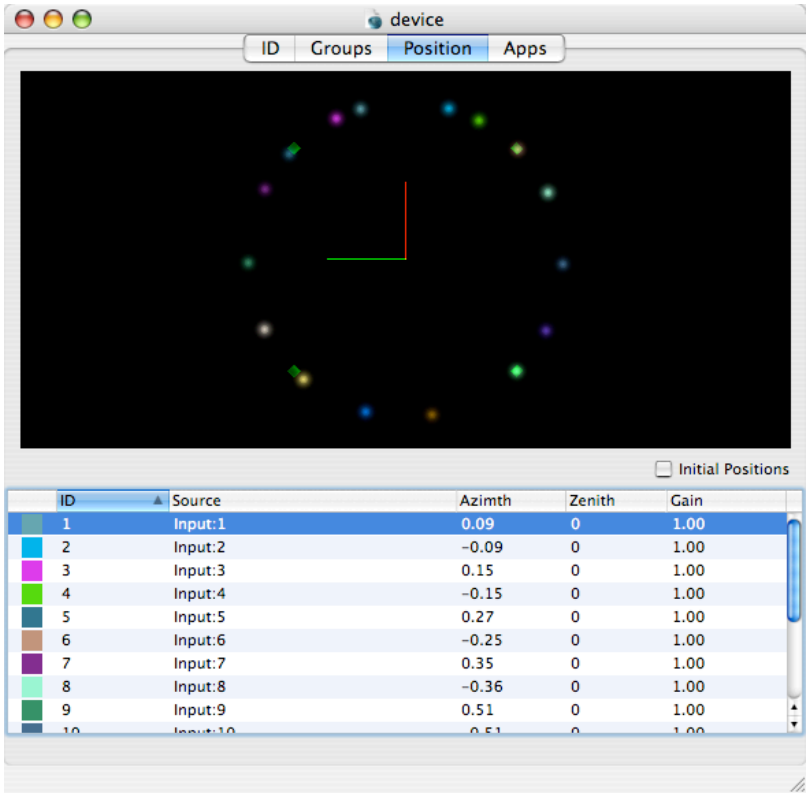
Groups have the same function within the context of a device that they do for pieces: they are a means of panning multiple IDs simultaneously.



Device Groups

POSITION

A device's IDs can be positioned just like a piece's. Though the IDs may be moved around, a preference file stored in ~/*Library/Application Support/Zirkonium/device.zrkdxml* remembers the initial positions of the IDs and restores them whenever Zirkonium is started. What differs from a piece is that there is save a sequence of movements because the device has no way of following the timeline of another program. It is still possible choreograph movements, but this must be done in the client program, either via the Zirkonium Zirk2 AudioUnit plug-in or via the OSC control interface. Moving the task of scheduling and invoking movements to the client



Device Position

program lets them be played synchronous to the timeline.

Spatialization View

Shows the positions of the IDs. IDs can be moved around within the view via the mouse.

Initial Positions

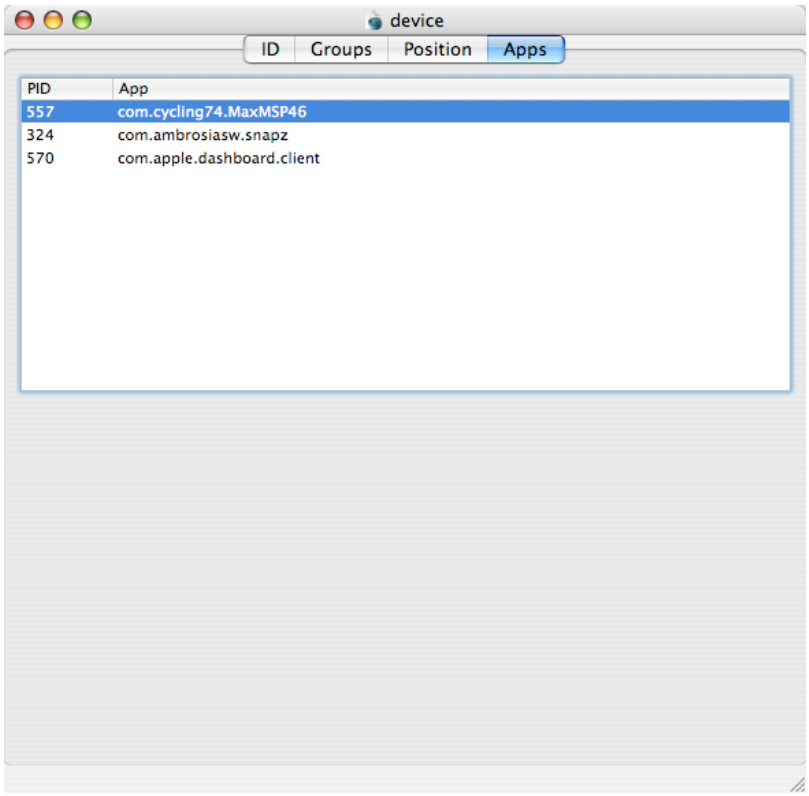
Toggles the display of the spatialization view between the initial positions and current positions. Initial positions are saved in the preference file and reinstated every time Zirkonium starts; current positions are not persisted.

ID Table

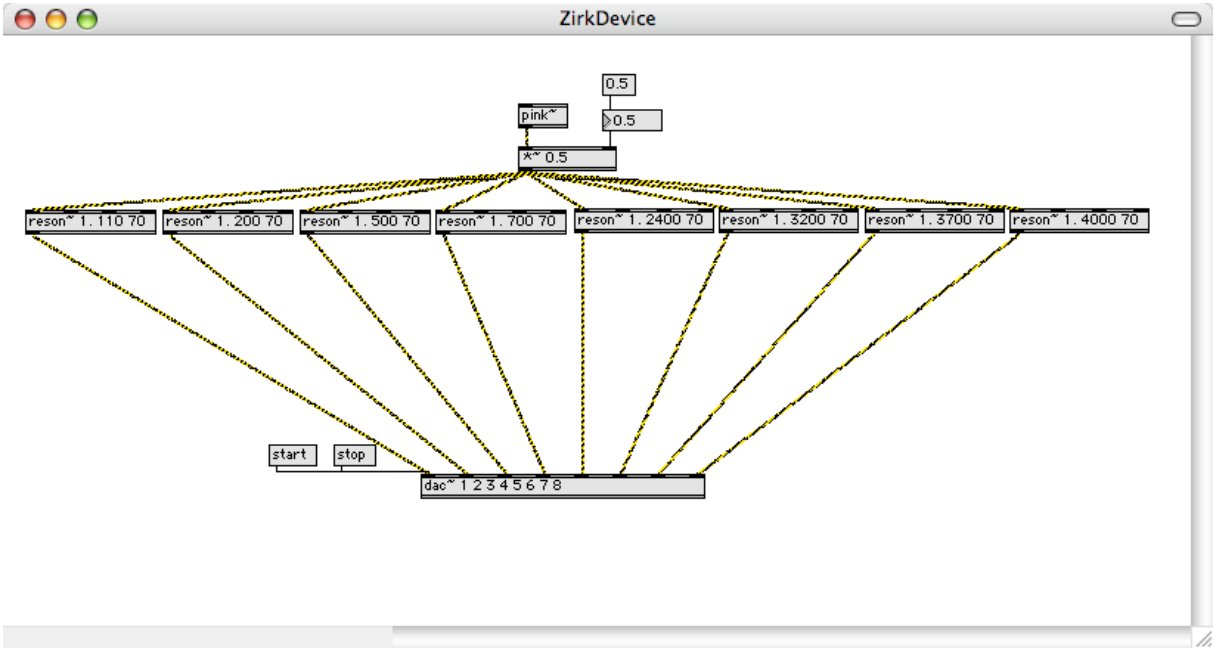
Displays the IDs along with their initial azimuth, zenith, and gain.

Apps

Displays a list of programs that can potentially connect to Zirkonium.



Device Apps



Using Zirkonium from MAX/MSP

USE

Using Zirkonium from another program is straightforward. The program must be configured to select Zirkionium as its output device. In some programs this can be done within the program itself. For example, in Max/MSP, the *dac~* object provides a dialog box for setting the output device. For other programs, such as SuperCollider, Zirkonium needs to be selected as the default input and output device in Audio MIDI Setup.

OSC Control

Zirkonium accepts control messages via the Open Sound Control (OSC) protocol. These commands are forwarded to the frontmost window, which may be either a piece or a device. (**NOTE: this does not currently work correctly — at the moment the command is forwarded to *one* of the windows. To ensure the correct window receives the command, close the other windows. This will be fixed in a future version.)** Messages should be addressed to */pan/az* with the arguments id, azimuth, zenith, azimuth span, zenith span, gain.

/pan/az (int, id) (float, azimuth) (float, zenith) (float, az span) (float, zn span) (float, gain)

Commands take effect immediately as they are received. In the future, we intend to add ways to address groups as well as IDs and add a */pan/xy* command that takes Cartesian coordinates instead of spherical. Until these options are implemented, the

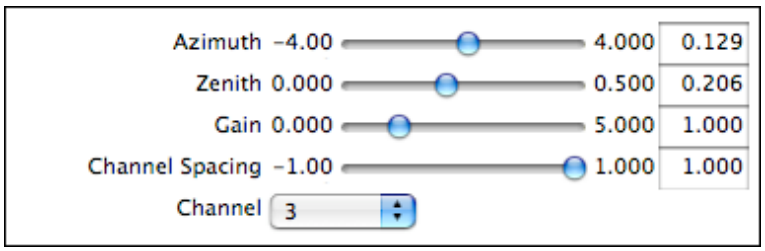
users of the OSC interface need to perform grouping and conversion from Cartesian to spherical coordinates themselves. For Max/MSP and SuperCollider, we provide objects that take care of these tasks.

A note about backwards compatibility: the previous version of Zirkonium accepted the message */pan/jump*. This continues to be supported with the first argument, the track, being discarded and the channel mapped to ID.

AudioUnit Control

OSC is a good means of controlling Zirkonium from programming environments like SuperCollider and Max/MSP, but digital audio workstations (DAW) do not usually provide generic OSC facilities. For DAWs we have developed the Zirk2 AudioUnit which sends control data to Zirkonium. To apply the audio unit to your your DAW project (e.g., in Logic or Digital Performer), refer to the manual for that program.

Audio Unit UI



Zirk2 AudioUnit UI

Azimuth

The azimuth of the selected ID (channel)

Zenith

The zenith of the selected ID.

Gain

The gain of the selected ID.

Channel

The ID this instance of the AudioUnit applies to.

FAQ

Q: Why are my OSC messages are not being received?

A: See the note in the section on OSC.

Q: Why doesn't the headphone simulation work with Max/MSP (SuperCollider, Logic, etc.)?

A: Yes, this is a known bug and will be fixed soon. The simulation does not currently work in device mode, and, in fact, connecting a device to Zirkonium will switch off the simulation.

Q: How can I use Zirkonium with a USB device or with the built-in audio on an Intel MacBook?

A: You need to create an Aggregate Device in Audio MIDI Setup that combines the input and output side of your USB device or built-in audio. See http://www.apple.com/pro/techniques/aggregateaudio/ for a step-by-step guide.

Q: Can I reduce the amount of CPU Zirkonium uses?

A: Yes! Having a Spatializer View visible (a view where the sound source positions are displayed) takes up considerably more CPU than having it not visible. Just switch the piece or device document to a tab without the visualizer, such as the *ID* tab.

Q: I'm running a piece and I hear the sound sources moving, but the display does not update. Why?

A: You are looking at the *Initial* tab which only shows the initial positions of the sound sources. Switch to the *Events* tab.

Q: I'm using Leopard, but can't get the device mode to work. What's the problem?

A: Update to the latest version of Zirkonium. The current version supports device mode under Leopard, whereas previous versions did not. Device mode is generally stable with most programs, though it may occasionally crash under some (e.g., Garage Band). Keep checking for updates as we are continuing on working to improve its stability.