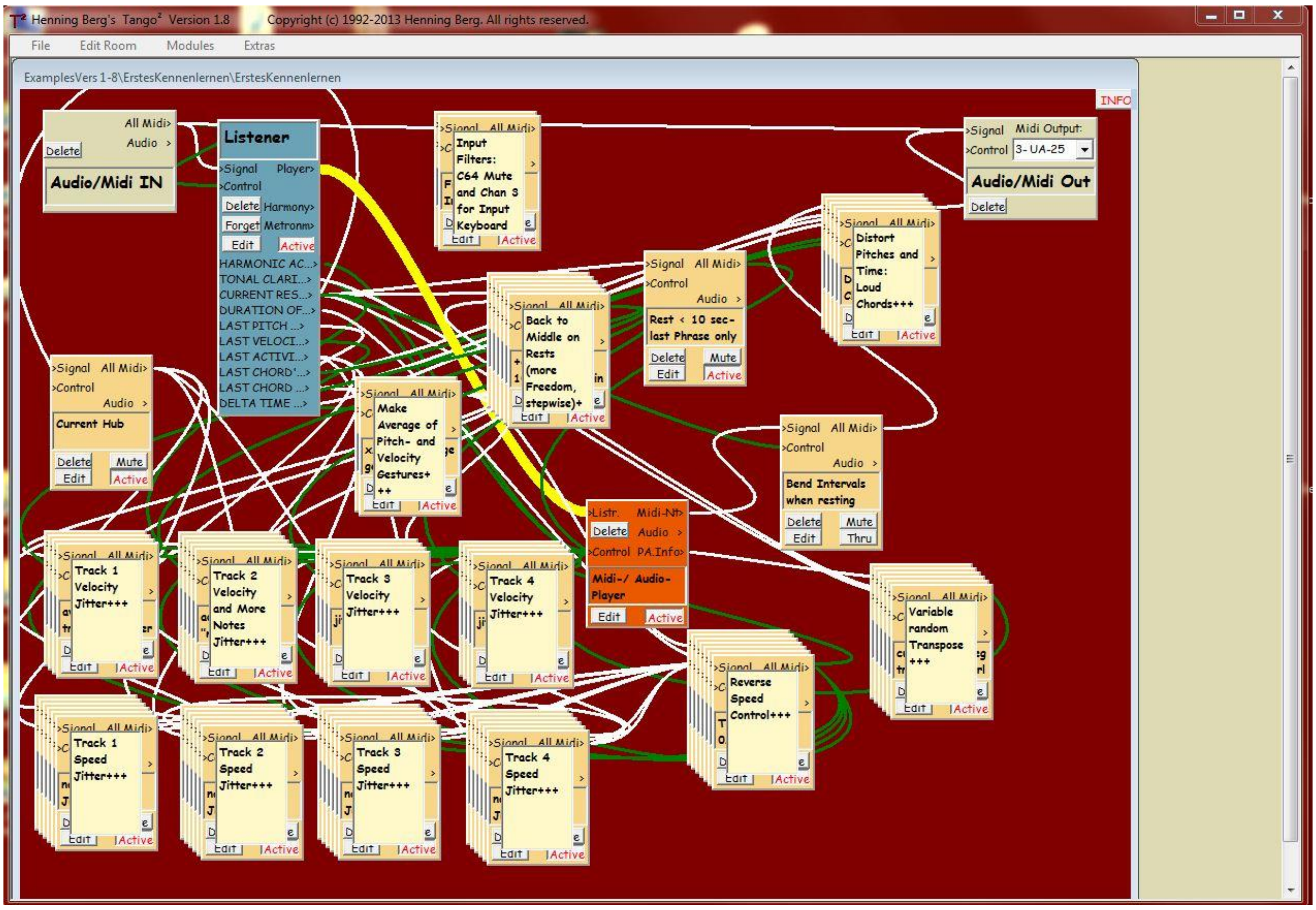
Henning Berg's TANGO²

Version 1.851

Manual

Index:


Henning Berg's Tango² Version 1.8
Copyright (c) 1992-2013 Henning Berg. All rights reserved.
File
Edit Room
Modules
Extras
ExamplesVers 1-8\ErstesKennenlernen\ErstesKennenlernen
Audio/Midi IN
Listener
Input Filters: C64 Mute and Chan 3 for Input Keyboard
Audio/Midi Out
Midi Output: 3-UA-25
Distort Pitches and Time: Loud Chords+++
Rest < 10 sec- last Phrase only
Back to Middle on Rests (more Freedom, stepwise)+
Current Hub
Make Average of Pitch- and Velocity Gestures+ ++
Bend Intervals when resting
Midi-/ Audio-Player
Track 1 Velocity Jitter+++
Track 2 Velocity and More Notes Jitter+++
Track 3 Velocity Jitter+++
Track 4 Velocity Jitter+++
Variable random Transpose +++
Reverse Speed Control+++
Track 1 Speed Jitter+++
Track 2 Speed Jitter+++
Track 3 Speed Jitter+++
Track 4 Speed Jitter+++

First Steps

If you are familiar with Midi systems (otherwise first read the chapter “Connections to the outside world”), connect a Midi keyboard to the Midi In- and a sound module to the Midi Out port of your computer. Coming from the keyboard, the Midi signal must go through your computer and Tango² before it reaches the sound module.

In the sound module select a piano sound on Midi channel 1. The keyboard should also send on Midi channel 1. An additional sound, running on Midi channel 3 will help you to differentiate your playing and that of Tango².

Your computer should run Windows7 or higher and you should use an ASIO audio interface. Mac Users please refer to the appendix of this manual for the installation under Boot Camp, which is explained step by step.
Please start Tango² and open the Room "ErstesKennenlernen.room" via the menu "File> Open Room".
Now your monitor should look similar to the image above. Do not try to understand the tangle of connections, after reading this manual this will be no problem for you.

Before you get started, you have to introduce Tango² to your setup. The connections for Midi In and Audio In / Out are set up in Extras> Global Settings. At the top left you can select your Midi In and your audio interface.

Finally you must look up the grey module "Audio/Midi Out“ at the top right of the Room display (the main window with the many connections). Below the words “Midi Output” you must select your ASIO interface from the menu, if it is not already selected.

That is the setup for now. If you like you can try out this Room by following my suggestions.

The basic idea for playing with T² is to enter an improvisational “Room”, make a musical statement then listen to what comes back from the Room. Not a long monologue, but rather a short statement like “Hello, anybody there?” works best at this stage.
What comes back will sound different from what you played, but will more or less be related to your input. If you now **react to that,** T² will hear and process this reaction – and so on. This is what the program was built for. In the course of your improvisation, the rests between your phrases will of course be shorter or longer (also much longer), however playing with Tango² will only be interesting if you listen to the program.

You are not an experienced pianist? All the better. First, use only the right hand, play monophonic phrases like a horn player, excluding any chords. Later you will of course use chords and even input audio signals instead of using a Midi-keyboard.

T² is not about playing complete pieces of music and then having the program “digest” them. Whole pieces are “ready” and this software is about proposing something unfinished as part of a larger “whole”. With Tango you should play things which could perhaps **be part** of a polyphonic piece rather than already delivering the complete accompaniment with the melodies.

Tango needs, at least in this Room, rests between your phrases. Long rests even. Take your hand off the keyboard during these rests instead of waiting for the next phrase with the last piano key pushed down (as this will “sound” like a very long note for T²).
I know that leaving rests of (e.g.) 5-10 seconds is somewhat harder for pianists than for horn-Players, but the program in this Room needs time to work with your ideas and to make something new of it.
You should **try to be inquisitive as** to what T² does with your ideas, how it layers them over each other, reduces/enlarges them or links them in other ways. You may hear an interesting permutation of one of your last two phrases and use this change in your next statement. This is much easier if you do not play in that moment of listening. Slowly a loop will emerge between your reactions to Tango and Tango’s reactions to you.

Be patient!
Often there is not much happening during the first 5 seconds, however then Tango may offer a good idea which coincidentally has to do with ideas of yours. That helps you then get into the “loop”.
In the long run you will develop a feeling for finding a balance between drifting through both Tango’s and your reactions on one hand and more targeted musical statements from you on the other to steer the program in a certain direction.
You are influencing the music strongly, as T² is imitating you in this Room, using and altering your phrases.

There are, however more direct effects of your playing: The (Listener-) parameters "Tonal Clarity" and "Harmonic Activity" as they describe this aspect of your playing, play a major role in this Room. If you play very “tonal” (e.g. triadic movement, open fifths), the program will mainly use your very last phrases (which were **tonal** – see above) instead of using older ones. So this is a way (in this Room, not with Tango in general) to capture T² in your tonality, making the overall music sound more tonal. Suddenly altering your playing from clear tonality into a completely different, atonal harmonic world can result in a wild outburst from the program, which eventually slowly calms down.
I built many other sources of variation into this Room using the modules you see on the screen like Lego bricks. You can open the Info-box in the right upper corner of the Room to find more information about this, but at this point I would advise otherwise.

You could start like this:
Play a short recognizable phrase of about 8-10 notes with the right hand. Let go of the keyboard and listen to what Tango does with it.
Wait a bit before you play again...
...
and a bit longer...
...
Now (maybe after 10 seconds of resting) play a phrase of a very different nature. After some time (...patience...) the phrases will be linked, connected or bent.
Now about a minute has passed and you have only played 16 notes...

Think from time to time of qualities such as tonal/atonal, loud/soft, fast/slow or high/low, especially if you want the joint improvisation of Tango² and yourself to go in a certain direction...

What does the program offer which you could incorporate or use?
Improvise with **that,** imitating or developing it and – again - listen to what the program comes up with after your last statement.

This is how the musical loop between you and Tango can work.

In theTango²- folders you will find a folder named "Musikbeispiele":
There are two recordings I made. One of them was recorded live exactly with this Room "ErstesKennenlernen.room".

Basic Ideas

On www.henning-berg.de you will find a lot of detailed information about basics and history of my Tango-project which doesn't need to be repeated here. You will also find a Windows executable version of Tango I.0, published 1990 by Steinberg for the Atari, including the manual and some additional texts, most of it in English.

Here is a quote from the website, which concerns Tango² (T²):

"With Tango² I try to incorporate my longtime concert- and programming experience (and that of many others Tango I users) in a software that concerns many of the old, basic ideas in a more flexible and comprehensive way.

The key new features of Tango² include

- *its clear modular design:*
  There are many different software devices (modules) which the user can put on the work-surface in any number or configuration and wire them up. Part of the program are Listener-, Player-, Modifier and other modules. They can be applied very flexibly to tailor simple or complex stimulus-response configurations according to one's own needs. Particularly important is also:

- *the integration of Midi and audio in all parts of the program:*
  T² has an integrated Audio- To-Midi module for monophonic instruments such as wind-instruments, voice and strings. This makes it possible to communicate with the program directly via a microphone (without a Midi-keyboard or an external Pitch-To-Midi-device). Additionally, T² can not only play Midi-synthesizers but also use and alter the audio-signal it received moments ago from the human partner for its reactions (and actions). The same modules are used for audio and Midi-functions, so it makes no difference to the operation whether Tango² works with Midi or with audio."

The user configures Rooms, i.e. environments, in which he will later improvise.
A concert with Tango² usually consists of a "Room" or a "Suite" of several rooms, which are called upon in order to improvisationally pass through or move within. For starters it might be helpful to open Rooms from the "Rooms and Tracks"-folder first and to change them to their needs to get a feeling for the program's functions, rather than starting from scratch.

T² can hold many Rooms in memory, similar to text processing software that can have multiple documents opened simultaneously. At any time however, only one room is active. There are also ways that make it possible to go through various defined playing situations with sequences of Rooms called Suites. Single Rooms and complete Suites can be saved or opened.

Comments on Rooms, also on the examples mentioned in this manual can be found via the "INFO" button in the right upper corner of most Rooms. There are several other places for INFO-buttons in the program, e.g. in the Player's and other modules' editors.

If "INFO" is written in red, an explanatory text is stored in it.

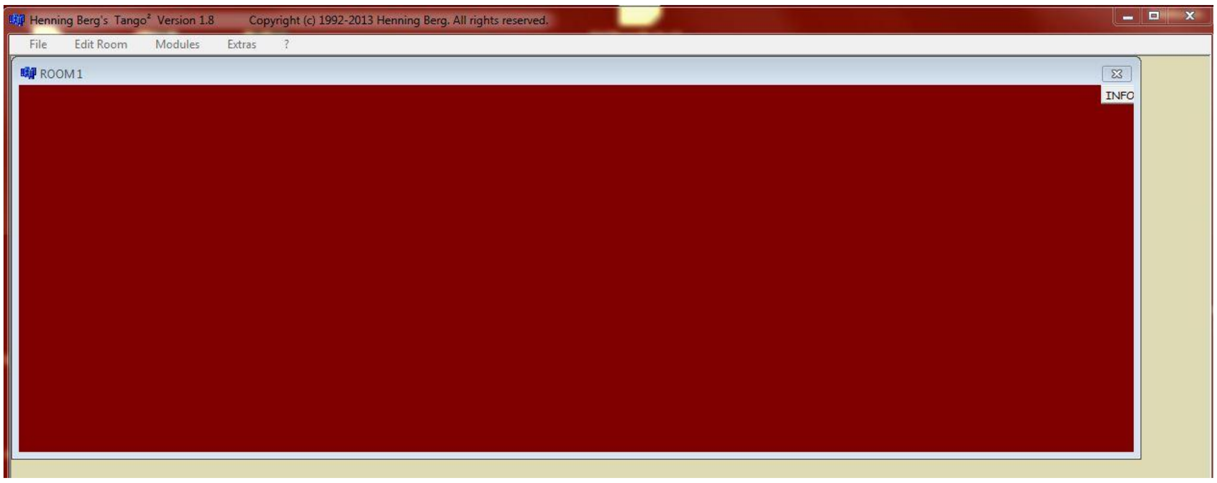
Tango^2 is not complete yet.

It looks completely different from the old program of the early 90s. It offers many new possibilities and quite well-developed and flexible modules. The Listener alone consists of 15,000 lines of code and writing this took almost 3 years; more than the entire Tango I (which also had a Listener).

On the other hand some key features of the old Tango I, like some of the harmony and rhythm modules do not yet exist for T^2. They will be available in the future.
With the current level however, it is already possible to experience exciting music with this program.

Connections to the Outside World

When the program starts you will see a red empty workspace, the room view, on which the User interface of T² is based.


Henning Berg's Tango² Version 1.8
Copyright (c) 1992-2013 Henning Berg. All rights reserved.
File
Edit Room
Modules
Extras
?
ROOM 1
INFO

To communicate with the outside world, you must open at least one Audio/Midi In- and one Audio /Midi Out-module in the Modules menu and connect them with each other or with other modules.

Your computer needs a Midi interface for Midi communication, firstly to play external Midi tone generators via Midi Out, and secondly to receive Midi-data via its Midi-In connector from a Midi keyboard.

The connections for Midi In and Audio In / Out are set up in Extras> Global Settings, while Midi Out can be configured in the Audio/Midi Out module. (The reason for this is that you can use multiple and differently configured Midi Out modules.)

ASIO is required for the audio handling. This protocol is currently the standard for all audio applications on the PC. Every serious audio interface (of which I know) supports it.

For the I/O functions the program offers links to the ASIO hardware that you have installed on your computer (Extras> Global Settings).

If you cannot use external audio hardware, I recommend the installation of ASIO4ALL, which supports built-in sound cards. However, then there is still need for a Midi solution.

The internal audio format cannot be selected in T². The program works with a sampling rate of 44,100 Hz and on the input side it is only listening to the left stereo channel. At the audio output, there is of course stereo.

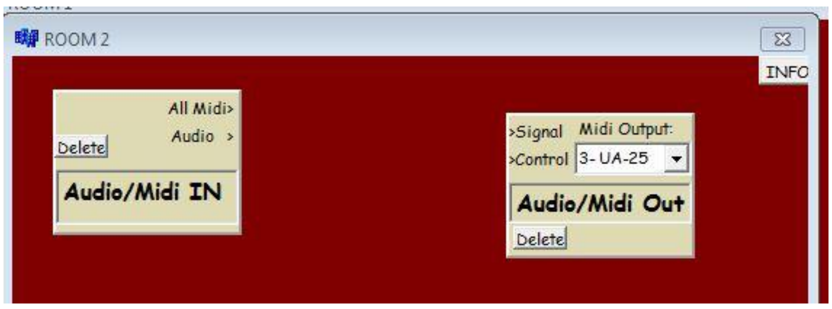
Combining Audio signals and Midi affords many interesting sound-possibilities.
Many audio interfaces also do the Midi-handling and have built-in Midi ports.

The Midi keyboard should send on channel 1 for the beginning and the signal should go “through” the PC and T² (and not directly to the tone generator).

Internal Wiring

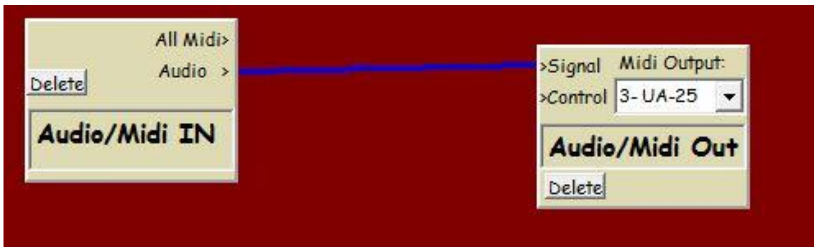
Each module has inputs and/or outputs which define the signal and control paths.
You will be drawing connections between them.
Because the In-module’s “inputs” are the physical inputs of the computer, it has only outputs to other T²- modules (and vice versa for the Out Module).
All other modules have both input and outputs.

Inputs and outputs are labeled according to their function. Inputs of a module are **always on the left,** outputs are attached **always to the right.** Any number of connections can go out of or end in a module. The few exceptions to this rule will be politely announced by the program.



Click on New Room in the File menu and select one Audio/Midi In- and one Audio/Midi Out module in the “Modules”- menu. You should now have an In and an Out module on the workspace. They should not overlap. Now click in the In module on the word "Audio", hold the left mouse button, move the mouse over the word "Signal" of the Out module and release the mouse there. If there is no blue line connecting the modules now, you may have used either the wrong mouse button, did not push it down exactly over "Audio", or you did not release it exactly over the word "Signal". In this case, repeat the procedure.

You now have produced a simple “Through”-connection between the audio input and the audio output of your system.



The modules can be moved by dragging the mouse and all the connections will be retained. To do this, click on the text box with the module name.

A connection from "Midi" to "Signal" forms a "Through"- connection for Midi data, but now it is white instead of blue.

Now, if the input signals are not audible, something is wrong with either your Windows sound system, your Midi setup, microphone or with your monitor system.

Inputs of modules marked with "Signal" can receive both Midi and audio data.

Connections can be deleted by holding down the Ctrl key while drawing them again (now with a red wire) and releasing the mouse in the input, just like before.

A tip for connections between modules:
For complex rooms, it is sometimes important to know what will happen first, for example if two connections are starting at the same output.
Here the order of drawing the connections is important as the commands will be executed in the same order by T².
If something does not work as expected, it may help to delete connections again and to draw them in the correct order.

The Inputs "Signal" or "Control" in the Modules

There are several types of information you can send from one module to another:

- **Signal data** are Midi notes, Midi controller (white) or audio data (blue), with which the receiver module should **do something.** If the receiving module were a radio, signal data would represent the electromagnetic waves.

  Control data (green) are used **for real-time control** of one or more parameters of the receiver module. Here you control the knobs and switches of the receiving module to define **how** the module does what it does. It works like a remote control.

The color of a connection provides information about its function. They are assigned by the program automatically.

- **Blue** is audio,

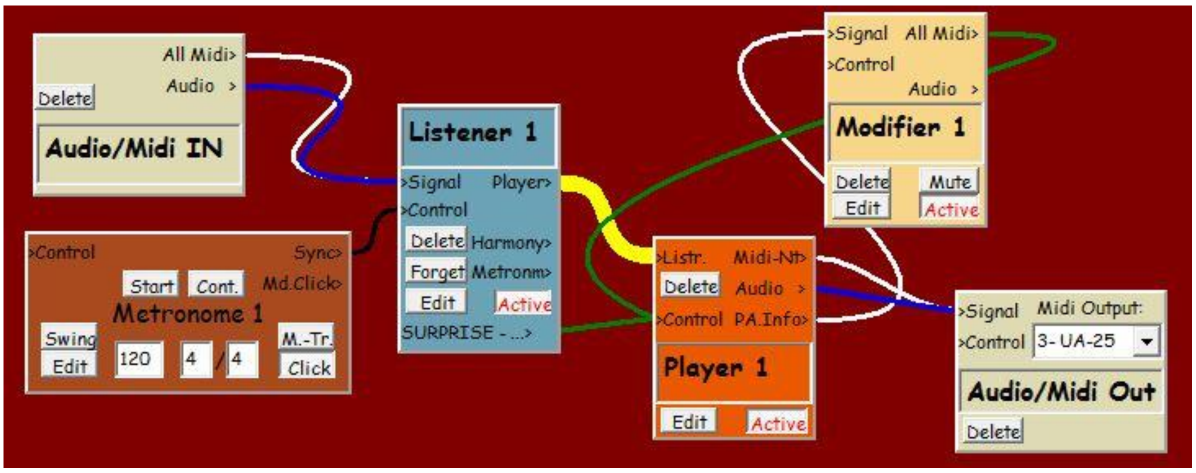
- **White** is Midi, going to a “Signal” input.

- **Green** is Midi that ends in a "Control" input of a module.

- **Yellow** is a connection between a Listener and a Player. This way the Player has complete access to the Listener’s evaluation parameters and its memory. Independently, individual connections from the Listener to
  Players can also be drawn. They are control messages, so they will appear green.

- **Black** are connections that have to do with time synchronization such as tempo.

In complex Rooms wiring the modules can quickly become confusing.
As long as you click on the name of a module all connections that are not related with the particular module will be hidden.

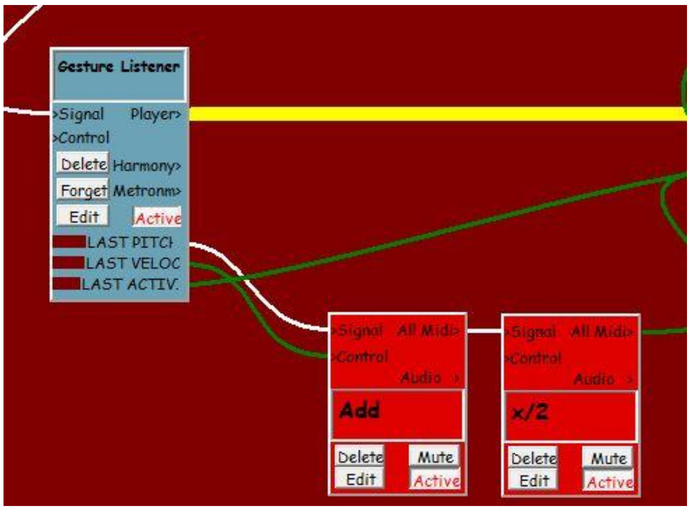


The Menus of Tango²

File

Open Room and Save Room as ... are the normal file functions for saving and opening Rooms.

Save Selected Modules
By double clicking on a module, it can be selected and deselected. Selected modules appear red in the Room view.
By simultaneously pressing the Ctrl-key and double-clicking several modules can be selected at once.



Various operations can then be used on the selected modules (here, only the selected Modules will be saved). The resulting file is still called "<name>.Room". It can be opened alone or added to existing Rooms. More in the menu section “Edit Room”.

New Room will generate a new, empty room.

Open Suite and
Save All Rooms as...

Suites are Multi-Room-Setups. They are important for concerts. Multi-Room-Setups can be controlled via the Room buttons on top of the main-display and can be remotely switched one after the other by the “double click” function (see “double click function” in this manual for that).

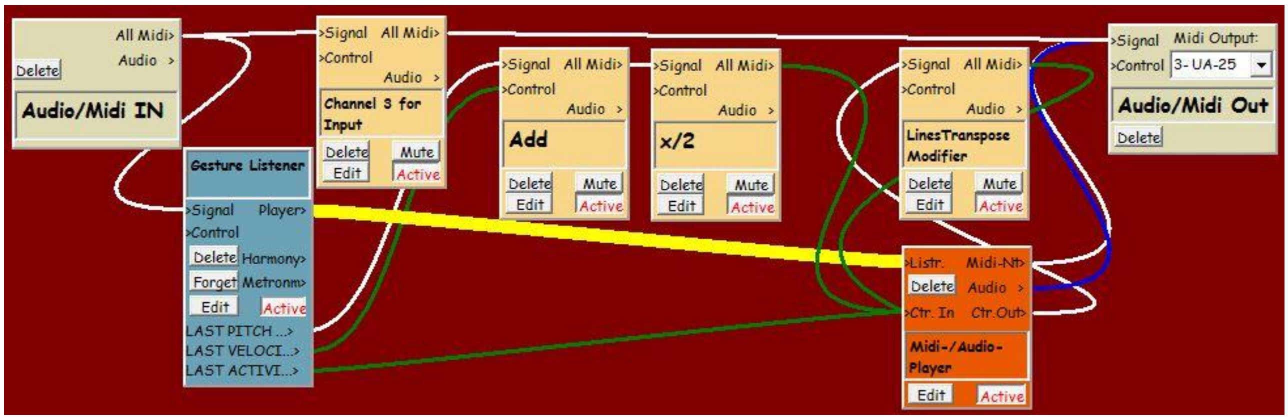
Edit Room

All commands in this menu refer only to modules in the active Room.

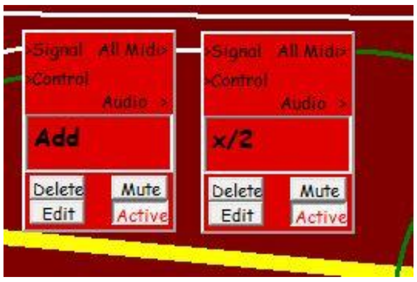
Copy/paste as usual offers this possibility for **selected Modules.** You can also copy them to different Rooms.

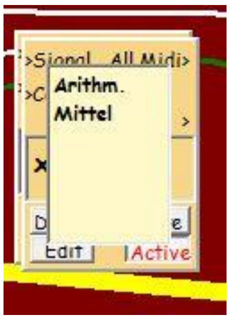
Create Group of Selected Modules

Complex Rooms are often composed of many modules, especially of many Modifiers. You have probably noticed this in the "ErstesKennenlernen.room". Many of these modules work as a group and fulfill a certain partial function in the Room while their group's internals are no longer interesting after the configuration. It is possible to group these modules together in order to make the room easier to understand. Some examples of such functionally related groups I have collected in the folder "Useful Macros". You will find explanations behind the Rooms' INFO button.



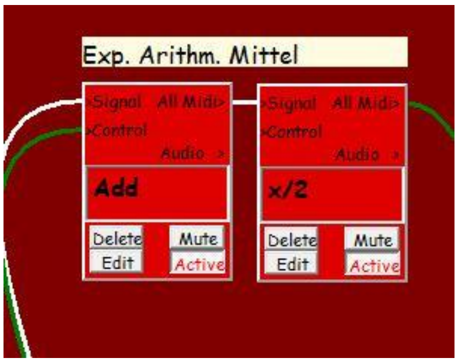
Open a new, empty Room with "New Room" and then open from the "Technical Demos" folder "ExampleGroups.room". The two Modifiers - what a Modifier is will be explained later - "Add" and "x / 2" calculate the arithmetic mean of the two values, so they have a have a common functionality. Select both successively with the Ctrl key and double-clicks. Now go to the menu "Create Group" and answer "Yes", as well as the next question about the label.





Now the two Modifiers appear stacked one behind the other, and on top one there is the possibility to enter a new name, e.g. "Arithmetic Mean ".

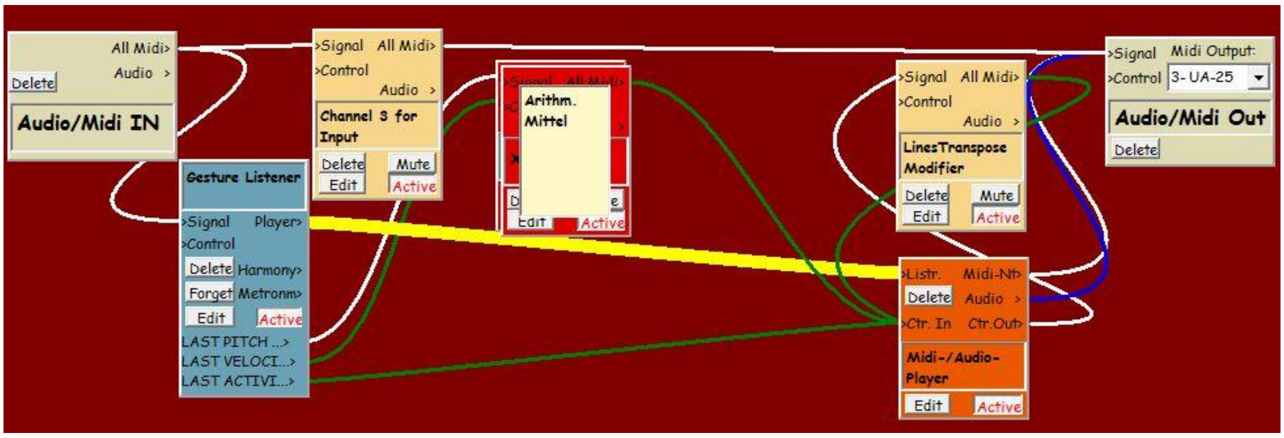
Expand Selected Group offers the possibility to display a group with several separate modules in order to change internal settings in the participating elements. The group status is not given up through this operation.



If the group "Arithm. Mean" is not displayed red (selected) anymore, reselect it by double-clicking. Now go into the menu to "Expand ..." and answer with Yes. You will see the modules involved under a beam with the name of the whole group, and now you are able to edit them with their Edit buttons individually.

If you want to remove a module from the group, this is possible by double-clicking on the module. Make the desired changes and return to the old - clustered - state again using

Restore View before last Expand.



After expansion, all modules which are involved are red.
You can remove modules from of the group by double-clicking to de-select them.
They will not be deleted by this operation – only removed from the group. If you remove the penultimate module of a group, logically the whole group will be disbanded.

The group handling is still a bit tedious. One of the next T²-versions will provide a display of groups as a unit, with normal-to-use input and output.
These new units can be grouped together again in a hierarchy.

Select / Deselect All selects all modules present in the active room (or deselects them if one or more were already selected).

Show/Hide Complete Wiring
The internal wiring of groups is normally hidden, unless you activate the last menu item in the Edit Room menu. Moreover, when clicking on its name with "Hide" active here, all connections that do not involve this module will be hidden.

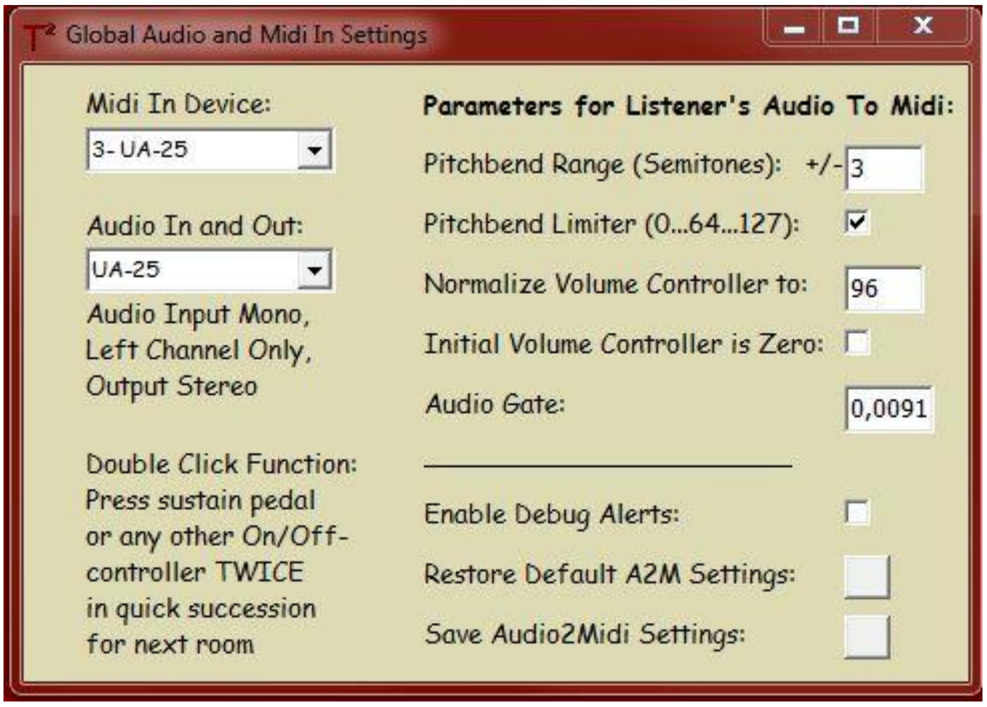
Extras

Global Audio and Midi In Settings

Here you configure the connections for Midi In and Audio In/Out, while Midi Out can be configured in the Out module. The reason for this is that several Midi Out modules with different configurations are possible.

Double Click Function

If you press any Midi controller like a sustain pedal (similar to a mouse button) twice in rapid succession, T² will move on to the next Room, if you have configured a "Suite" of Rooms. This is another important function to avoid obvious clicking and switching at the computer during concerts.



T² Global Audio and Midi In Settings

Midi In Device: 3- UA-25

Audio In and Out: UA-25

Audio Input Mono, Left Channel Only, Output Stereo

Double Click Function: Press sustain pedal or any other On/Off-controller TWICE in quick succession for next room

Parameters for Listener's Audio To Midi:

Pitchbend Range (Semitones): +/- 3

Pitchbend Limiter (0...64...127): ✓

Normalize Volume Controller to: 96

Initial Volume Controller is Zero: ☐

Audio Gate: 0,0091

Enable Debug Alerts: ☐

Restore Default A2M Settings:

Save Audio2Midi Settings:

On the right you find parameters for the Audio-to-Midi functions. They are discussed in the section Audio To Midi (A2M - see below).

You can **Save Settings** you have made here for Audio To Midi.
These are automatically loaded at program start
Also, you can return to my (H.B.'s) preferred default settings.

Editing Numeric Fields

Throughout the program numbers can be specified after clicking by typing them as usual. You can also drag the mouse after clicking and keeping the left mouse mutton pressed to increment or decrement values.
A confirmation with the return key is usually not necessary, so the values are effective immediately.

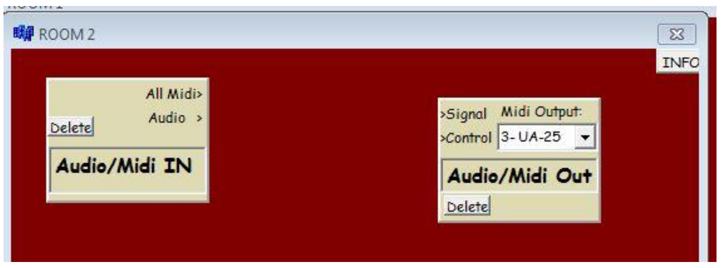
The Description of the Currently Existing Modules

All modules except "Audio/Midi In" can appear multiple times in a Room.

Audio/ Midi In

is directly connected with the audio and Midi input devices of your system. Here the signal enters the world of T². You can only call one “In” for each room. The configuration is done in Extras> Global Audio and Midi In Settings.

Audio/Midi Out



is Tango’s connection to the existing Windows Outputs for Midi and audio. Here the signal leaves T². Midi Out is configured here in the module as there may be several different Midi Out modules in one Room addressing different physical outputs.
All other configurations can be found in Extras> Global Audio and Midi In Settings.

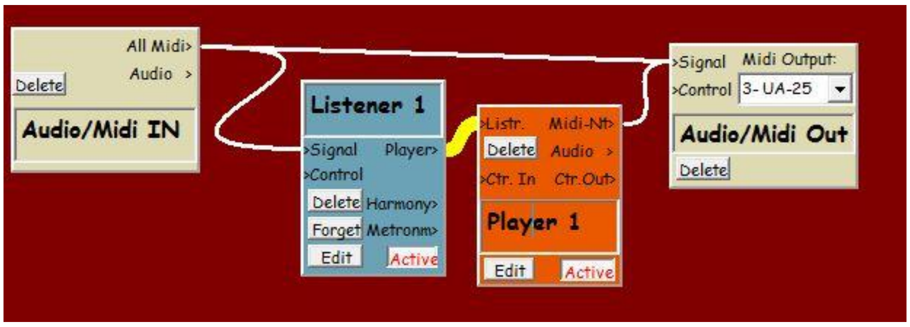
Player

If you improvise with Tango², the music the program plays usually comes from a Player module. It produces the musical answers to your playing according to certain rules,

This "raw material" of Tango's output can be adjusted in the course of the signal path with various Modifiers to meet your needs about dynamics, harmony, meter and rhythm etc. The notes however, are usually produced here in the Player.
To try out the features of the Player, please open the following modules in an empty room ("New Room ")

- Audio/Midi In,
- Audio/Midi Out,
- Listener,
- Player

Arrange them so the room looks similar to the following image.
The modules can be moved by clicking in their name box and dragging the mouse. Names can be changed with the "delete-" or the "backspace" keys.



Wire up the modules by first clicking to "All Midi" in Audio/Midi In and moving the mouse (left key down) to "Signal" in the Listener. The Listener's "Player" output must be connected to the Player's "Listener"-input and Player's "Midi-Nt." out to "Signal" in Audio/Midi Out.
Now the only thing missing is a direct link from "All Midi" of the In- module to "Signal" of Audio/Midi Out, which is a simple Midi-Through-connection. Now you can not only hear Tango playing, but also yourself.

Connections are drawn by clicking in the output, holding down the left mouse button and dragging the mouse to the desired input. Release the mouse there. T² assigns the colors automatically.

If you play a short phrase on the Midi keyboard and everything is wired correctly, T² will continue to repeat this phrase quite monotonously for the next few years…
You can stop that by clicking on the Listener's "Forget" button.
If all works up to this point, you should save the Room with "File>Save Room as ..." because we will return to this later. As a name I suggest BasisPlayer.room.

Your playing is the musical information to which the Player module responds. It receives this information via the yellow connection from the Listener, which listens to you, remembers and analyzes your playing.

The Player has direct access to the Listener's memory and the results of its analysis, which is called "Evaluation" in T².

The Player learns what the Listener "knows" about your playing based on the Listener's memory. The white lines show you that this Room is only for Midi information on both the listening and the playing sides. By adding blue lines to the audio outputs (can be found under the Midi-
outputs), you can easily extend the system for audio.
For clarity, we will stick to Midi for now.

The link of your playing with some response of T² is provided by a **Player algorithm (abbreviated PA).** Here the rules are selected and modified according to how T² reacts to your playing.

The Player can play its answers in Midi-notes or in samples of the sounds that the program has just heard from you. For working with samples it is a prerequisite that you use a monophonic instrument (wind-, string instrument or voice) and a microphone for the input instead of a Midi keyboard.

In the future there will be many different Player algorithms such as "**Values",** which freely generates completely new phrases from the results of the Listener's evaluation, a **Drummer / Sequencer** that is able to alter given sequencer tracks in real time or "C**ues",** which reacts to your playing with Crescendi / Decrescendi or/and Accelerandi / Ritardandi. Those familiar with Tango I will find many similarities between the new Player and its algorithms and the former "Main Modules".

Lines
The first completed Player algorithm (PA) is Lines, which can produce a polyphonic texture of your previous phrases.

For Lines "phrases" are groups of notes with a pause before and after the group. You can refine Tango's definition of a phrase in the Listener more accurately. But for now, let's be satisfied with my preset values.

The Basic Function of the Player

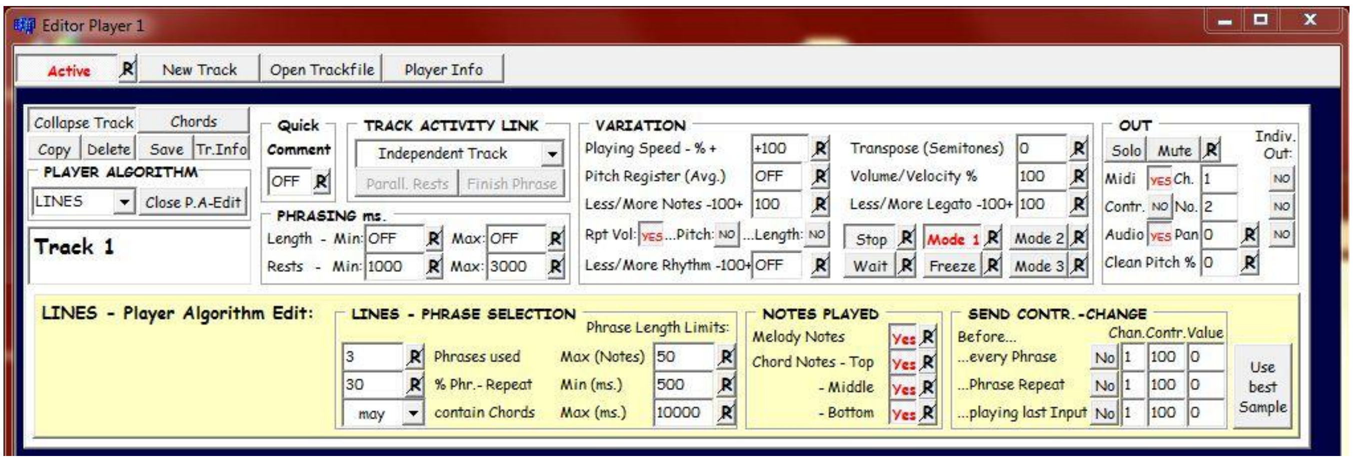
The Player can contain one or more (up to 200) tracks that, using the Listener-Data, either simultaneously or alternately produce music. This can happen in different ways, but at present, as I said, the only PA available is Lines.

You will notice though that Lines can react and even act in a very flexible way.

Each track alone can produce monophonic or polyphonic material, including chords.
Multiple tracks can be activated, so that Lines itself also can produce polyphony and chords, even if the human partner has played only monophonically. Melodies can also be wrapped in chords by the Player.
For details, see the section on the "Chords" button and under Player-Chords.

Player Edit

Open the Player editor in the newly created Room "BasisPlayer.room" by clicking on the Edit button in the Player module. Now you can see the Player Editor with one track.



PLAYER ALGORITHM: LINES
Track 1
Quick Comment: OFF
TRACK ACTIVITY LINK: Independent Track
PHRASING ms.
Length - Min: OFF Max: OFF
Rests - Min: 1000 Max: 3000

VARIATION
Playing Speed - % +: +100
Pitch Register (Avg.): OFF
Less/More Notes -100+: 100
Rpt Vol: YES ...Pitch: NO ...Length: NO
Less/More Rhythm -100+: OFF
Transpose (Semitones): 0
Volume/Velocity %: 100
Less/More Legato -100+: 100

OUT
Midi: YES Ch. 1 — Indiv. Out: NO
Contr.: NO No. 2 — NO
Audio: YES Pan 0 — NO
Clean Pitch %: 0

LINES - Player Algorithm Edit:

LINES - PHRASE SELECTION
3 Phrases used
30 % Phr.- Repeat
may contain Chords
Phrase Length Limits:
Max (Notes): 50
Min (ms.): 500
Max (ms.): 10000

NOTES PLAYED
Melody Notes: Yes
Chord Notes - Top: Yes
- Middle: Yes
- Bottom: Yes

SEND CONTR.-CHANGE
Before... Chan. Contr. Value
...every Phrase: No 1 100 0
...Phrase Repeat: No 1 100 0
...playing last Input: No 1 100 0

On top of the Player editor you will find some **Basic functions** such as the possibility to

- **deactivate** the Player completely (this is corresponding to the same button on the Player module of the T² main window, the Room-display)
- **open** a new track or
- **open** a **previously saved track** ("<Filename>. Track") and
- hold **Info** about the current configuration of the Player.

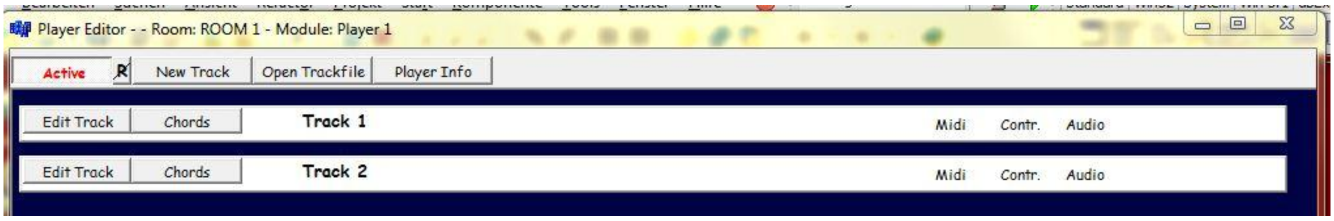
Track-Representation in the Player Editor

Each track box is divided into two or more differently colored areas.
In the white area on top, the parameters of a track are arranged. Here, a track's way of playing can be adjusted or modulated in real-time**, regardless of the choice of the PA.**

In the light yellow area below you can define how the selected PA is supposed to work.
This area therefore will look different, depending on the selected PA**.**

On the left side you will find a group of buttons:

Collapse Track offers the possibility of displaying a track clearly and narrowly, if no adjustments have to be made.



"Chords" opens the chord editor of the track. In the default setting there are three new voices that form chords, together with the main-voice according to certain rules. This number of voices can be changed between 2 and 20. For more, see the section "Player Chords".

Click again on the Chords button and you will be asked if you want the chords-function for this track turned off or only switch to the narrower view;

Right now you want to turn it off completely.

Copy creates a new track with identical settings.

Delete deletes this track.

Save saves an individual (namely this) track, which can be opened via "Open Track File" and imported (e.g.) into another Player ("<Filename>.track").

Track Info allows for note-taking on this track.

Below you will be able to choose a **Player algorithm (PA)** in the future and on the right next to that, the light yellow **PA-editor** can be closed or opened.

Finally, there is the possibility to give the track a new **name.**

The Track Parameters

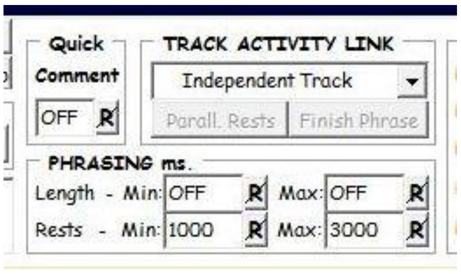
From left to right in the white track edit box:

Quick Comment

With this function the track can play a quick and relatively short response to things you have been playing very recently , after making a longer pause (of at least 1.5 seconds). This "comment" will only use the material you have played right **after** this rest and accelerate/slow it down and/or transpose it on several layers.
The amount and length of Quick Comment is highly dependent on **what** you have played in this last phrase. In addition, you can determine the amount and the length of QC in two ways:

- by setting a Quick Comment value here in the editor.
- by configuring a Real Time Control (RTC)-connection so that Quick Comment is controlled in real time by the Listener. For details on RTC, please look below for Variation> Playing speed. There is also an example.

If you want to listen to Quick Comment alone in order to see how the function works, click on the right of the editor on the button "Stop". The PA (Lines) is not working now, but Quick Comment, which is independent of the PA, does. After rests of yours, depending on the set value for QC, Quick Comment will pick up and die down again after some time, until the next Quick Comment comes along.



Phrasing

Under the Quick Comment box you find settings that are important for the length of phrases and the pauses as T² plays them - always in relation to this track.

Generally, a phrase is a group of notes with a rest before and after.

If you enter the same millisecond values for Min and Max, the track will produce phrases and rests of this length. If the maximum value is greater than Min, the value will be determined randomly within the given range.

With the phrasing parameters you can configure a track so that it forms shorter or longer phrases than those you played. Lines finds your phrases in the Listener's memory.
"Off", which is also can be selected, means that your phrase lengths are used unchanged. How Lines selects and uses phrases is explained below (Lines - Player Edit Algorithm).

Track Activity Link

Since phrase-lengths and pauses between phrases can be defined, it is clear that tracks play sometimes and sometimes not (namely, where they rest).
This change of activity of tracks may overlap and occur randomly when multiple tracks are involved, but it may also be synchronized.

Add a new track or copy an existing track, then play a few phrases of a few seconds in length. Both tracks play and pause independently.

Now you can choose from the Track Activity Link menu of track 2 **"Follows Track 1".** Both tracks will play together and take their breaks together. Here track 1 is master and track 2 the slave.
Under this selection, you can choose between **Parallel Rests** and **Alternating Rests.**

If track 1 is playing, track 2 depending on this setting will rest or play. Track 1 will act independently on its phrases and breaks, while track 2 follows. If track 1 wants to "end a break" (that is play the next phrase), you can use
Finish Phrase or **Phrase Cutoff** to decide whether track 2 finishes its phrase first or cuts off the phrase that it is playing right now.

Variation

Here you will find the track parameters which customize the playing of a track to your needs as well as changing it at short notice.

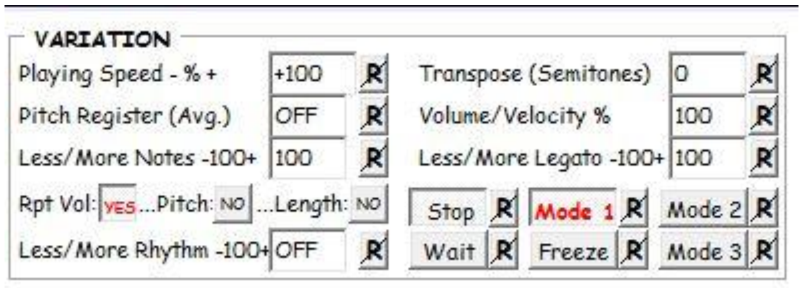
Playing Speed is the speed with which the track plays. Possible values are up to 1000% as well as negative values. At 100% Lines plays your phrases in the original tempo, and negative Values of course mean playing backwards. At 0% there is a standstill.

Pitch Register (Average) defines a Midi pitch.
This parameter changes the pitch of this track in such a way, **that the average pitch of its playing is as close to the value given here as possible**. Transposition happens only by octaves, so that the pitch classes (C, D etc.) remain unchanged.

In the Midi protocol middle C (the center of the keyboard) is defined as pitch 60, and each + / - 12 semitones signify a transposition of an octave up or down.

It is therefore possible to quickly change the register of a track (e.g. for a bass track with the value of 60 – 12 = 48), yet keep the harmonic content (the absolute pitches such as C, C #) intact. "Off" again means that no change is desired.

Less / More Notes adds notes to the music the track is playing (values above 100%), or takes notes away (under 100%). Notes are taken away from phrases by making the notes that were played before longer. So here no additional pauses are produced and the phrases are not shorter or longer than before.



VARIATION

Playing Speed - % +	+100	Transpose (Semitones)	0	
Pitch Register (Avg.)	OFF	Volume/Velocity %	100	
Less/More Notes -100+	100	Less/More Legato -100+	100	
Rpt Vol: YES ...Pitch: NO ...Length: NO		Stop	Mode 1	Mode 2
Less/More Rhythm -100+	OFF	Wait	Freeze	Mode 3

Which and how many notes are changed, is randomly determined based on the %-value.
When notes are added (values above 100%) notes of the phrase will be shortened for other short notes to be inserted before the original note. In the next line can be determined whether these new notes have all the same ("Repeat ...") or different pitches, volumes or lengths as the shortened original note. “No Repeat” for the volume signifies a slight crescendo of the new notes until the original volume is reached. Pitches and durations are varied randomly.

Less / More Rhythm makes the rhythm, i.e. the quarter notes, eights, sixteenths etc. of phrases more varied or more uniform. Positive values change the phrase so that the length of each note varies more, while negative values will lead to more even note values as continuous quarters, eighths or sixteenths.

OFF means as always, "no change". With “OFF” in Lines the rhythms of your phrases remain unchanged. Other changes, such as the Speed parameter or Less / More Notes of course still are possible and remain valid.

Transpose changes every pitch by a number of semitones (+ / -). The operation is different from Pitch Register. Here we simply transpose, thus "C #" with setting of -1 becomes a "C".

Volume / Velocity % changes the volume (in the Midi protocol called velocity - the speed with which a key is pressed down) of the tracks relative to the original volume.

Less / More Legato changes the ratio of notes and rests between them. The track leaves the Legato quality of the original phrases unchanged with a setting of 100. A value of 200 signifies tightest Legato, so there is no pause between consecutive notes. A value of 1 means the shortest possible Staccato of each note with a subsequent pause until the next note starts. This parameter does not affect the rhythm, i.e. the note beginnings of a phrase.

The **Stop / Start** button activates / deactivates the Player algorithm of this track. As stated above, despite the "Stop" Quick Comment still can generate notes.

Wait / Continue stops the track while notes that are still running are being held until "Continue".

Freeze causes the track to just keep playing as it is at present. So Lines keeps using the same phrase, and the entire RTC system of the track (explained in the next section) is blocked.

The three Mode-buttons, finally allow saving and recalling all parameters of a track as one of three presets with a push of a button (or a real-time control command). To save the presets use the same button and hold Ctrl. At this point a preset is only stored in the computer’s memory. To save it permanently you must use track-, module- or Room-Save-operations.

Once you select another “Mode”, the previously active mode is deactivated and replaced by the selected one, so you can only select, but not deselect Modes.
Modes include the RTC system (Real Time Control) of all parameters, which will be described in the following section.
If RTC is activated on a parameter, the Mode will **not store the value** of the parameter, but the **RTC parameters that enable its real-time control,** i.e. the Midi channel and controller number with which it is controllable instead. If Chord parameters in the track are active, the Chord Settings are treated like any other track- parameter. Excluded from the Modes are here "Number of Voices" and "Main Track Position in Chord“ plus all parameters of the individual chord voices. You will find more on the Chord system below.

1st Interlude: The Application of Midi-Controllers to Control Parameters in Real Time and the Use of Tables (RTC)

*Here is a short detour on **Real Time Control (RTC)** using the example of the parameter "Playing Speed"**:***

Already in Tango I, it was possible to influence Tango's music by your own playing- as you would expect it from an improvising duo partner.

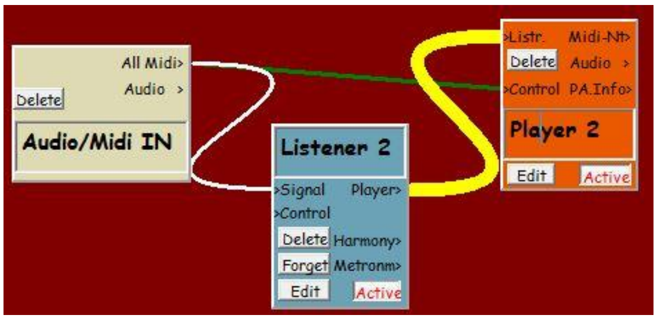
A given Player Algorithm linked input and output, and changes in my playing consequently led to changes in the playing of Tango.

*The new system of Midi controllers and Tables, in which parameters of a Listener module can change one or more parameters of a different (for example, Player-) module in real time now gives us the possibility to **constantly alter the reaction algorithm,** instead of just selecting an initial configuration (that is before the actual playing takes place).*

Now you are able to alter and manipulate the responding system during improvisation by influencing the rules of the game (instead of just playing it).

This is why the Listener has become so complex. This way the music will become less static and allows for more and longer developments and variations. It is important that this is possible without handling the mouse on the computer during a concert, simply by correlating certain properties of my music with specific characteristics of Tango's reactions.

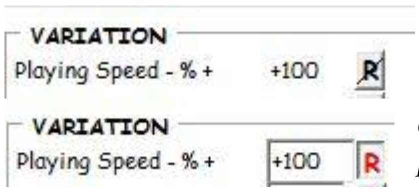
For our example, we will get the Midi controller directly from the modulation wheel of the keyboard as the functions of the Listener, which would normally do this job, will be described only further below.



Please ensure that the modulation wheel of your keyboard sends Midi controller 1 on Midi channel 1.

In our BasisPlayer.room connect "All Midi" (Audio/ Midi In) to the "Control" input of the Player. It is no problem if you have to draw over the Listener to do this. You should now see a green line between the two modules.

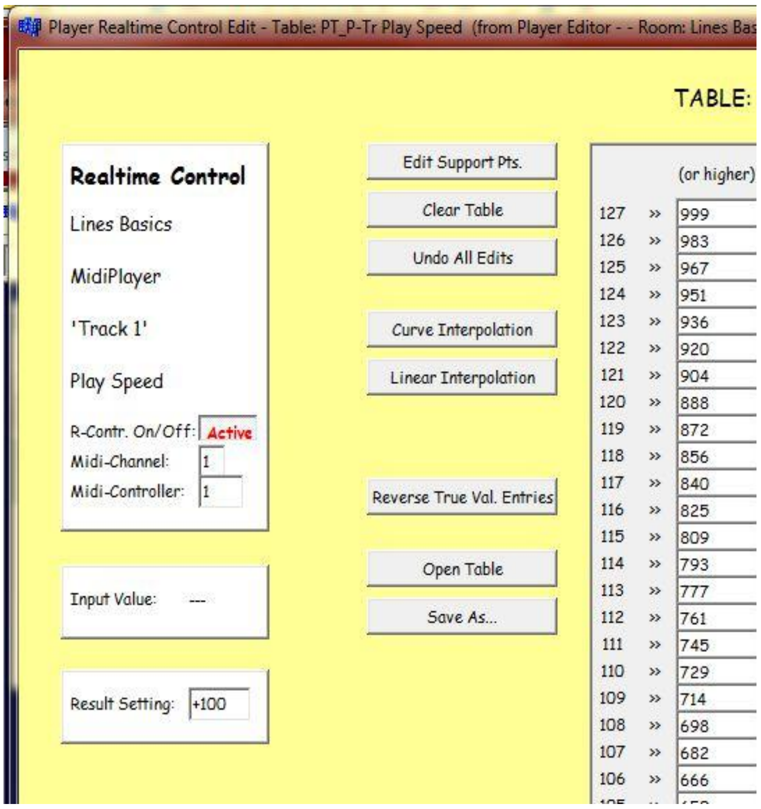
Click the Player's Edit button, so that the editor moves to the foreground.



To the right of the parameters in the Player editor, you will usually see a button with an struck-through R. It serves to activate the real-time control system of the respective parameter. If you click the button, two things will happen:

- *Real-time control is enabled for that parameter, the parameter becomes externally controllable, and the "R" - Button is not struck-through anymore.*

- *The Editor **Real Time Control Edit** for the parameter opens. In this dialog you find in the middle and right columns a list of numbers called "Table" and associated graphics and Edit buttons.*
  *How and why to work with Tables and how the **Table Editor** exactly works, will be explained right after this.*
  In the left column you can find information about Room, Player, the track and the parameters in this track that we are currently dealing with, (here Playing Speed).
  Next you find out here, whether Real Time Control is active and with which controller and Midi channel the parameters can be controlled.
  The "Active" button always shows the same activation state as the real time button ("R") next to the parameter (here Playing Speed) in the Player editor.
  The lower two panels show the value that reaches the Player via the green connection from the outside (usually it ranges from 0 to 127 – "Input Value") and ("Result Setting") the resulting value after the scaling from the Table.



Player Realtime Control Edit - Table: PT_P-Tr Play Speed (from Player Editor - - Room: Lines Bas

TABLE:

Realtime Control
Lines Basics
MidiPlayer
'Track 1'
Play Speed
R-Contr. On/Off: Active
Midi-Channel: 1
Midi-Controller: 1

Input Value: ---

Result Setting: +100

		(or higher)
127	»	999
126	»	983
125	»	967
124	»	951
123	»	936
122	»	920
121	»	904
120	»	888
119	»	872
118	»	856
117	»	840
116	»	825
115	»	809
114	»	793
113	»	777
112	»	761
111	»	745
110	»	729
109	»	714
108	»	698
107	»	682
106	»	666

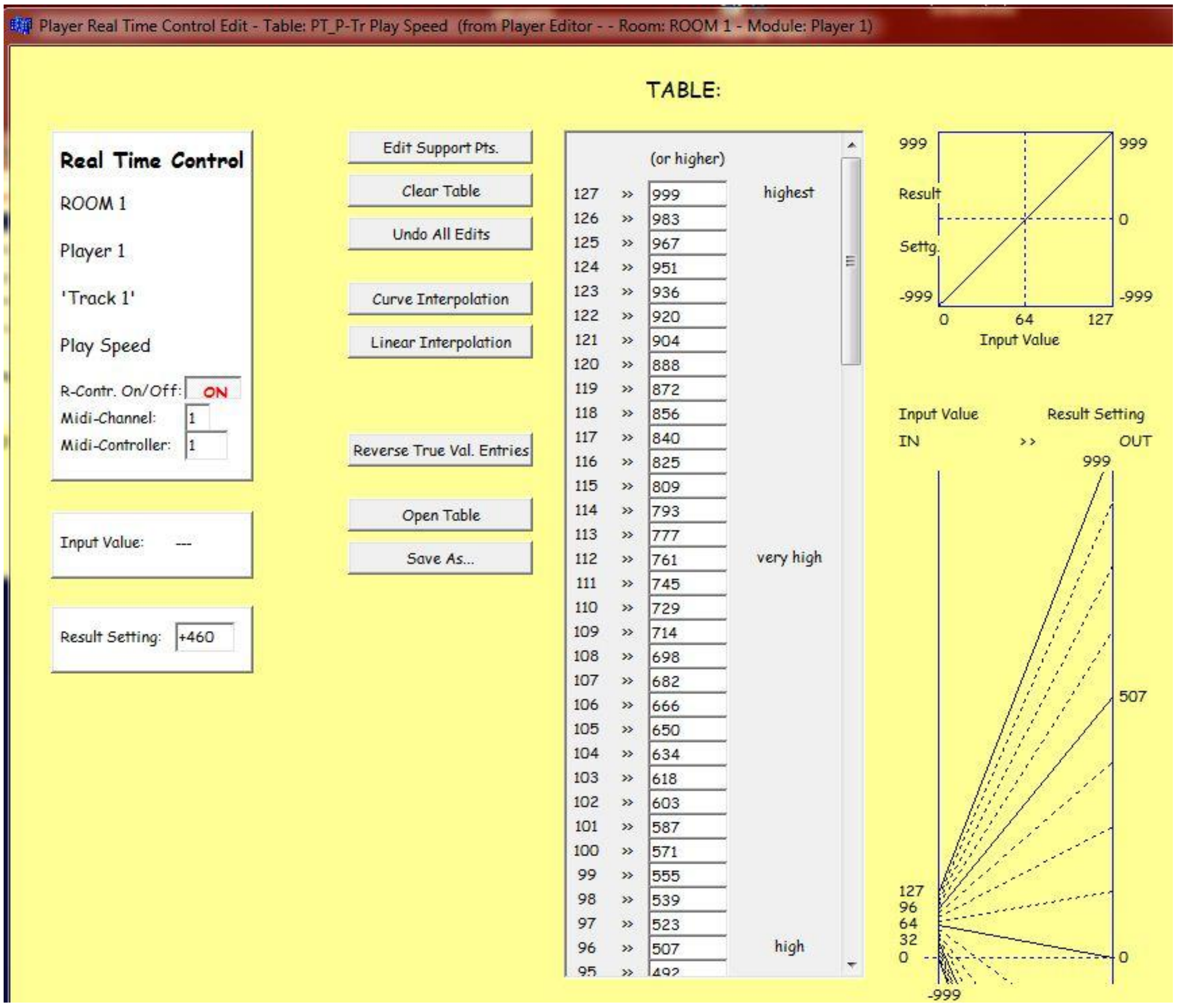
Now you can use the modulation wheel to control the speed at which the Player plays back your phrases. The range is 10 times the original speed forward and backward. You can follow this control on the Player's edit page in the Playing Speed box and on the RTC edit page under Input Value or Result Setting.

Controls of this type you will find in many examples about the Player, where you can test configuration, operation and effects. However, as I already mentioned, the source of the controller usually is not located outside the program like our modulation wheel.
Instead, it will often be a Listener parameter that describes one particular aspect of your playing which (after some scaling) controls certain aspects of Tango's playing.
If you are in the Room view, the parameters you have configured for Real Time Control are visible as a tool tip when you move the mouse over the control input of the Player module.

The Use of Tables
An important tool to enable such control functions are Tables.
Technically a modulation wheel puts out numerical values in the range between 0 and 127.
For our parameter Playing Speed (measured as a percentage of the original speed - forward

and backward) this is not particularly useful and therefore values can be converted/scaled using Tables.

Click (when needed twice) on the R button next to Playing Speed until the yellow RTC editor, with the Table of the speed parameter can be seen again.



Player Real Time Control Edit - Table: PT_P-Tr Play Speed (from Player Editor - - Room: ROOM 1 - Module: Player 1)

Real Time Control

ROOM 1

Player 1

'Track 1'

Play Speed

R-Contr. On/Off:	ON
Midi-Channel:	1
Midi-Controller:	1
Input Value:	---
Result Setting:	+460

TABLE:

		(or higher)	
127	»	999	highest
126	»	983	
125	»	967	
124	»	951	
123	»	936	
122	»	920	
121	»	904	
120	»	888	
119	»	872	
118	»	856	
117	»	840	
116	»	825	
115	»	809	
114	»	793	
113	»	777	
112	»	761	very high
111	»	745	
110	»	729	
109	»	714	
108	»	698	
107	»	682	
106	»	666	
105	»	650	
104	»	634	
103	»	618	
102	»	603	
101	»	587	
100	»	571	
99	»	555	
98	»	539	
97	»	523	
96	»	507	high
95	»	492	

The Table Editor

In the graph on the lower right below in the Table Edit window, you can see that values from 0 to 127, as they come from the modulation wheel (left side – Input Value) are converted into the right-hand values from -999 to +999. In addition you will find the same conversion represented as X / Y curve (the horizontal
X-axis here describes the raw values) and finally in the middle the editable number assignments shown in plain text. The values as they come from the modulation wheel are in the left column, right next to that you find the values available after the conversion /scaling.

"Scaled" and "True" Values
In the real world parameters have the range of values that they have - volumes

in western music range between "ppp" and "fff" or sometimes "ffff", Midi-modulation wheels send between 0 and 127, and Tango's Playback Speeds lie between -999 and +999 percent.

Many of these parameters, such as the length of notes (10 to 10000 milliseconds) and intervals (the value 12, one octave, in melodies is already a fairly large value) run in completely different value ranges and are therefore not compatible as raw values. Tables serve as a means to unify all these parameters, allowing to modulate a parameter with another. They are lists that allow the conversion to a scaled "standard parameter" for compatibility.

For T² a scaled value is one with a range as it comes from a Midi modulation wheel. Zero stands for "nothing", 32 is "low" and 80 means "well, already higher than the center (64), but still not really high (hence "highish") and 127 is the highest possible value. The colloquial descriptions of the height value, in addition to the numeric fields in the Table are listed in the editor.

Editing Numeric Fields
Throughout the program numbers can be specified after clicking by typing them as usual. You can also drag the mouse after clicking and holding to change values. A confirmation with the return key is usually not necessary, so the values are effective immediately.

Edit functions of the Table editor:

Clear Table
clears all values, except for the upper (+999) and the lower (-999) values. These two fields can be edited like the others, but they must have some value. Click on Clear Table and write in the top field (to the right of the 127) the number 300 instead of 999. Now scroll all the way down and enter in the Table next to the (left side-) "0" instead of -999 the value +20.
Also enter the number 80 on the right of the (left side-) number 96. At point 96 / 80 in the diagram on the right you will find a red circle that marks the point as a support point. Click on the button

Linear interpolation
Now a connecting line is generated which is fairly flat up to point 96 >> 80 to rise steeply beyond that point. If you now move the modulation wheel and the listen to the Player, you will notice that the playback speed which is in the lower three-quarters of the wheel's range is not changed much, but then quickly rises to 300%. Reverse playing does not happen anymore because you just eliminated all negative values from the Table.
Edit Support Points
Now again you see the familiar point 96 / 80 Change this value to (e.g.) 149 and additionally enter the value 82 for field number 64 ("Center"). Now click on

Curve interpolation *to obtain a curve, without the edges of the linear interpolation. This function works with up to two support points and tells you if it could not reach certain interpolation points. You can also switch between the two ways of interpolation.*

Undo All Edits *restores the original state, as it existed on opening the Table. There is no "Undo" Undo.*

If a Table converts a scaled "0 to 127- value" to a "true" value of the real world (such as here in the playback speed), you see "0 to 127" - values on the left in the Table edit page and on the right those of the real world. The flow of information is (as always in the program) from left to right.
Where useful, as sometimes in the Modifier, you can ***Switch Conversion Mode*** *manually to change to the opposite mode (with real world values on the left). Often the conversion goes first from real world values to 0-127 (e.g. if the Listener analyses note lengths) and then from 0-127 back to real world (if you want to control some Player parameter with this information). Most of the time, T²-Tables will provide the correct direction automatically.*

If this Operation is not useful, you won't find the appropriate button.
Use ***Reverse True Val Entries*** *to reverse the lowest and highest true-values of the Table.*

Open *and* ***Save*** *are the necessary open and save operations for Tables.*

Each RTC compatible parameter has its own Table.
There are three different places where Tables can be stored:

- ***Right here in the module:*** *The Table of the parameter "Playing speed" in the module Player is already in memory. It will be saved together with the module and all other Tables of the current Room's modules when the Room is saved. You don't need to worry about backing up edited Tables.*

- ***In the folder "Tables"*** *which is located at the same file-level as T². When the program starts, this entire folder is automatically copied into memory.*
  Here the Tables are stored, which get loaded into the module if you retrieve it from the T² main page via the Modules-menu. These are the default Tables.
  An edited Table only needs to be saved in the folder "Tables" if you want this edited version as a default Table for certain parameters. Otherwise, saving the Room (or module / track) will also save all the Tables involved.
  Tables are uniquely identified by a combination of letters and numbers (starting with "PT ..." for parameter Table). This enables the modules to find their appropriate Tables. After this initial labeling you are free to rename Tables, but the default Tables have to have their original names.

- ***The folder "Original Table Protected"*** *(found in the "Tables"-folder) is read-only and contains the Tables, as they were configured in the "factory"-version of the program.*
  A final backup to make sure that each module always finds the right Tables for the correct parameter.
  If you accidentally delete a necessary Table from the Tables folder, T² automatically looks here for the right Table in this backup folder.

More Control Functions

Double Click Function

If you press any Midi controller like a sustain pedal (similar to a mouse button) twice in rapid succession, T² will move on to the next Room, if you have configured a "Suite" of Rooms. This is another important function to avoid too obvious clicking and switching at the computer when you are playing live.

Pitch Switches

If you press a Midi-controller 64 (e.g. a sustain pedal) and play certain pitches or pitch-zones at the same time through your microphone or on a Midi-instrument, you can switch any RTC-function within T² this way. ***Use "Pitchswitch.room" as an example. In the "Info" of that Room you will find details on the configuration.*** *You will have to configure your Pitch Switches on your own.*

The point is again to be able to play concerts without using the mouse or having to step on multiple foot switches. Too much "public switching" often becomes a distraction for you (playing your instrument) and for the audience (listening to the music).

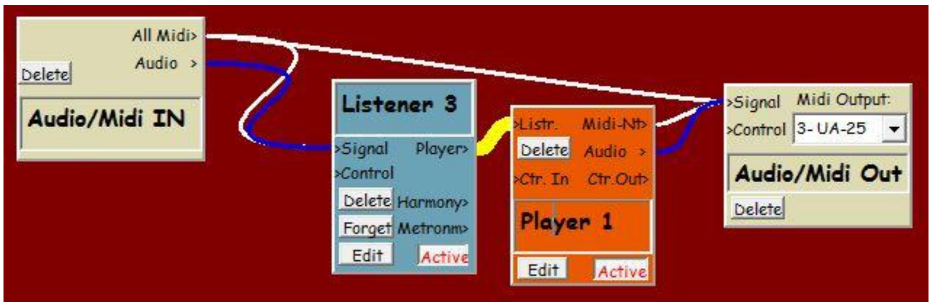
With the Pitch-Switches you can control sub rooms, harmonic or rhythmical elements, switch modules to "off" or "through" etc. in a very discreet way. For this purpose I use a small button on my trombone.

End of the description of RTC, the Real Time Control System (1st Interlude).

2nd Interlude: The Player and Audio Signals

Preparation of BasisPlayer.room for Audio Operation

Drag a connection from the Audio output of the In-module to the Listener's Signal input, and another from the Player's Audio output to the Signal input of the Out module. Your setup should look like this image.



Now connect your microphone to the left microphone input of your audio hardware and sing something into the microphone.
You should hear Midi and Audio played in unison by the Player.
(As a reminder: You can delete existing connections by drawing them again while holding down the Ctrl key.)

If it does not work:

- *Do the microphone signals arrive in T²? Are the microphone level controls OK? You can see the signal at the output of the audio-in module. Clip means you are too loud.*
- *Is the audio connection between your computer's audio hardware and your monitoring system OK?*
- *Do the blue connections match with the above image?*
- *Is audio switched On in the Player editor's "Out"- section?*

General Information for Dealing with Audio Signals in T²:

Audio signals and Midi signals are internally represented as single notes. Tango's Audio To Midi Converter (A2M) separates the individual notes from an audio stream, describes them with a resolution of 10 milliseconds in terms of volume and pitch and finally stores them separately in the Listener's memory, along with their descriptions.

The Audio To Midi module can be called separately, but it is also integrated in the Listener and therefore does normally not appear in the Room. A (blue) audio connection from the audio output of the In-module to the Signal input of the Listener does the job.

Audio To Midi works only with monophonic signals *like horns, voice or single note piano.*

This system enables the Player to use previously heard audio material for its musical responses by reshaping it, combining notes to chords or having them changed by other modules. So T² can not only use pre-produced synth or sampled sounds, but can also use your own sound to play with you. Of course, the Player can respond to your audio signals as usual in the "language Midi" or combine Midi and audio sounds.

As Tango's A2M, as I said before, only processes monophonic material, overlapping notes, such as playing very legato on a piano or un-muted guitar strings as well as thick reverb or delays can disturb the pitch tracking. Ideal are dry horn and vocal signals. Even with the guitar, I would prefer a Midi solution.

Setting correct levels for audio material as high as possible without clipping, improves pitch tracking. You can check the audio output of the in-module for that. I use a little compression before the audio interface because the trombone produces rather big variations between soft and loud playing. Saxophones are dynamically easier to handle, but they are difficult to record with a close microphone which helps avoiding "Listener-feedback".

On the other end, you should hear the ***output of Tango²*** *(the Player's output) as quietly as possible in your monitor, so that you are still able to react to the program but avoid the above mentioned Listener-feedback of Tango's music blasting from the speakers into your microphone (that is into Tango's Listener). Alternatively, use headphones. Feedback loops can provide a very interesting application of T², but they should happen inside the program and not between microphone and speakers.*

Only the left of the two stereo channels gets analyzed.

A2M represents each audio note with note-on and note-off Midi messages about its initial volume (velocity) and pitch.
Later (after the note-on and before the note off) the volume is further described with controller 2 (often called "Breath Control") and the changes in pitch with pitch bend.

The pitch and volume description happens at a resolution of 1/100 second (and only if something has changed in pitch or volume), this information is stored in context with the samples in the Listener's memory.

The Listener gets the audio events about 0.37 seconds after you have played them. This value can be different because of the potentially different audio latency of your hardware. Most of the delay however, has to do with the operation of my Audio To Midi module, and due to technical reasons, therefore cannot be switched off.

For T² this is early enough and it allows for good accuracy in tracking.

The pitch bend range is preset to + / - 3 semitones by me. You can change that value in the "Global Audio Settings" of the Extras menu.
All glissandi in the audio with a distance of more than three semitones from the original pitch will therefore not be described in this setting. If you want to control a Midi- tone generator with this pitch bend data, you should have the same pitch bend range adjusted there so that the result sounds like the audio original.

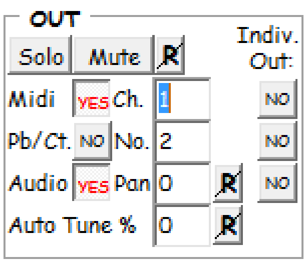
The audio detection is looking for usable ***pitches.*** *If you sing along and it use text, you will mainly hear the vowels in your singing, because only these contain recognizable pitches.*

So much for now about the audio operation of T². (End of the 2nd Interlude)
This and other information can also be found in the description of the
Audio To Midi module and the Listener.

Now back to the parameters of the Player Editor:

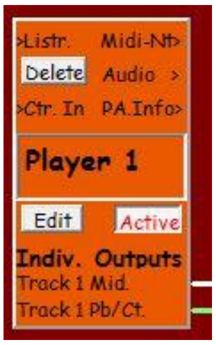
Player-Out

This determines which data (audio, Midi, pitch bend and controllers) the track sends, which Midi channel or controller it uses, and whether the (global) Midi -Nt> or a single output will be used by the Player.



Solo and **Mute** silence this (Mute) or all other tracks except for this (Solo). Mute is also RTC controllable, Solo is used only during Player configuration, so there is no R button.

In the “Midi” line you can decide whether the **notes** of a track are output as Midi data, which Midi channel is used and whether it goes to a **single output** (Individ. Out) or the Player’s general "Midi- Nt." output.
Individual outputs can be useful if for example, only the notes of one particular track should be processed separately by a Modifier or another module (e.g. the bass should play only roots and fifths of the chord changes).



If by clicking the appropriate button below Indiv(idual) Out(puts) individual outputs are configured, that signal does not appear in the "Midi-Nt." jack of the Player anymore. Instead a single output is attached in the Room display at the bottom of the Player icon. The wiring works as usual from other outputs.

Pb / Ct (pitch bend / controller) refers to the possibility of describing notes not only at their beginning by pitch and volume, but also in their further course by pitch bend and a volume controller. The prerequisite for this function is that the Listener received audio, so you must use a microphone for playing. This audio data is memorized by the Listener. Via the Listener's internal Audio To Midi converter it is then marked with the appropriate Midi information about pitch- and volume-envelopes of each note (in a resolution of ten milliseconds).
This information can now be sent along with the raw Midi notes by the track.
Pitch bend is the format for the pitch description.
With the "Pb / Ct" button both data streams (pitch bend and volume controller) are switched on and off.
With this it is possible to use fine pitch and volume ornaments such as vibrato or Crescendos that you have played in the course of an audio note for real-time control of a Midi sound in order to make it as vivid as your original audio signal.
Even oddities like crescendos on individual piano notes or a fine brass vibrato from the harpsichord are possible with this function. This requires that the corresponding sound in your Midi tone generator can "listen" to these controllers.

The **Controller Number** for the Midi volume is selectable and to the right, a single output for this data can be configured. If no single output is set, the controller output of the Player module appears in the "Midi-Nt." output as they are part of the notes.
Individual outputs allow for separate output of notes on the one hand and the associated pitch bend / controller data on the other.

A Midi-Tip: The pitch bend range T² uses to analyze the audio notes can be set in the

"Global Audio Settings" in the Extras menu. It should be the same as with your selected tone generator sound. Also the sound should not have the volume controlled by Midi velocity but by the selected volume controller. Find out more about this in the chapter on Audio To Midi.

Audio
If you have recorded a monophonic audio signal via a microphone, the track can also use the sounds of this audio signal for its output. This possibility can be configured in the lower two rows of the Out area.

Pan determines the position of the audio signal's left-right panning.

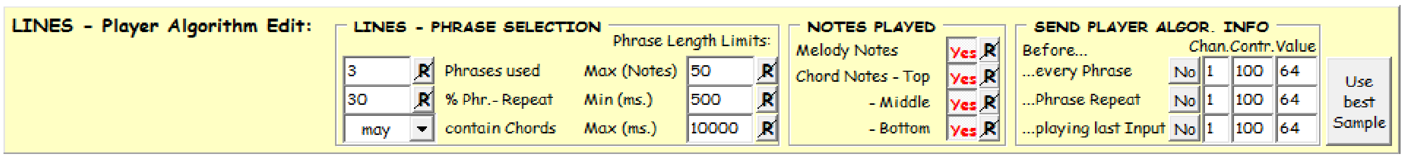
Auto Tune % can compensate for variations in pitch of your notes, so at 100% all notes are played pitch-corrected. 0% Auto Tune leaves the pitch unchanged. And on the very right you will find the single output for audio signals.

If you have recorded only Midi notes and no audio signal, the audio range as well as the above-mentioned Midi controller for Player notes is not available.

If under the white track editor there is no bright yellow region titled

Lines - Player Algorithm Edit, click on the Open-PA Edit button.

These parameters are only important for the currently selected Player algorithm (PA).
Other PAs will have different parameters, but for now they do not exist yet.



LINES - Player Algorithm Edit:

LINES - PHRASE SELECTION

Phrase Length Limits:

3 R Phrases used — Max (Notes) 50 R
30 R % Phr.- Repeat — Min (ms.) 500 R
may contain Chords — Max (ms.) 10000 R

NOTES PLAYED

Melody Notes Yes R
Chord Notes - Top Yes R
- Middle Yes R
- Bottom Yes R

SEND PLAYER ALGOR. INFO

Before...		Chan.	Contr.	Value
...every Phrase	No	1	100	64
...Phrase Repeat	No	1	100	64
...playing last Input	No	1	100	64

Lines has the following parameters:

Phrase Selection
Here you decide how Lines selects the phrase to play next from the phrases it finds in memory.
Generally, only the last phrases you played are taken into account.

Phrases used
The value of "3" means for example that the last 3 of your phrases can be used. Which phrase is actually used, will be randomly decided.
However, Tango's freedom can be limited by the following parameters:
% Phr.-repeat is the probability in percent that the most recently selected phrase will be repeated. If you have enabled only the very last phrase ("1 phrase used "), this value is meaningless. This parameter allows for a greater motivic coherence in the music of a track.

Your input phrases may have been monophonic or they included chords:
Here you can set whether phrases selected by Lines **must** contain chords or not, or if you do not care: Therefore, the options "must", "must not" or
"'may contain chords".

Lengths of Selectable Phrases are defined by a maximum number of notes or a maximum / minimum number of milliseconds. Phrases that do not meet at least one of these requirements will not be used by Lines.

Once a phrase has been selected (according to these rules), under

Notes Played you can decide which notes of the selected phrase should actually be played:

Melody Notes were **not** part of chords. Chord notes can be

Top (top note),
Bottom (lowest note) and
Middle (all other notes).

If you have selected Melody Notes and Bottom chord notes, all notes that were not part of chords and the lowest note of chords, the bass note, will be played.

The Listener of T² hears a chord, if at least 2 notes are played with a time interval of less than a certain number of milliseconds between them. How many milliseconds, can be defined in the Listener Settings under "Max Delta time between Chord Notes“. *For details, see “Chord Activity”, right before the section on Special Listener parameters.*

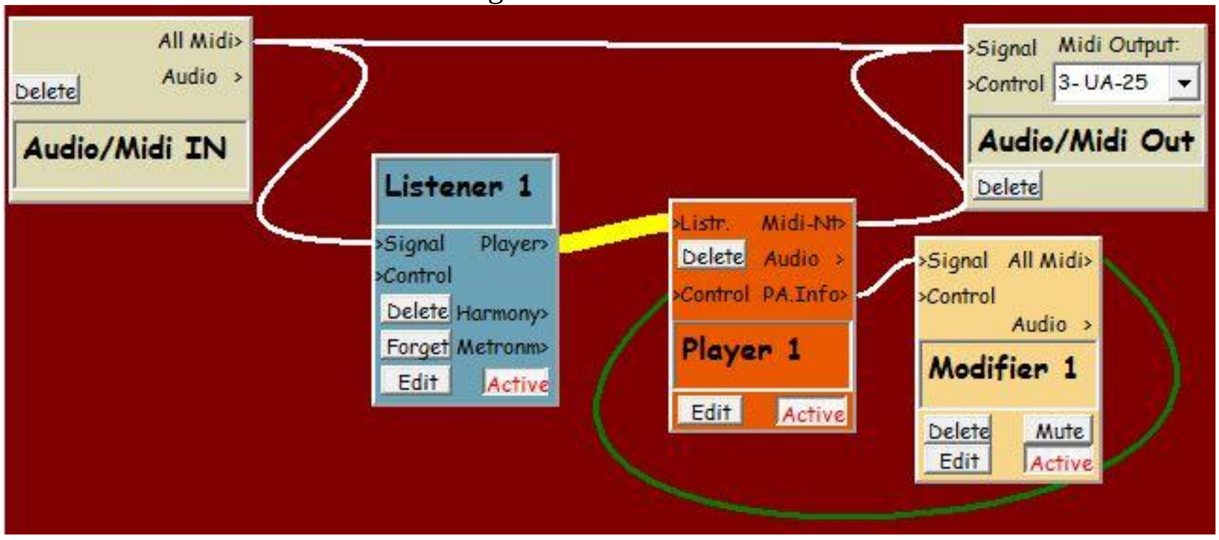
Send Player Algor(ithm) Info
Lines can inform other modules or (in a feedback loop) itself if it is about to play a new phrase. This can happen
before each phrase,
before a Phrase Repeat and
before playing the last input phrase (before it is about to use the last notes you just played).

For this P.A.Info, the channel, controller and value to be passed, can be defined on the right. Here controller numbers can be selected only from 100 up so they cannot interfere with controllers which describe Midi notes.

In order to keep them separate, these controllers always appear at the Player’s PA.Info> output and not at the Midi-Nt> or single outputs, if you have activated any.

An example for this function is the configuration of a "Lines Transpose Modifier".
Again, use BasisPlayer.room as a starting point, open the Player Editor and click on the button right next to "... every phrase" in PA Edit area. If
Lines wants to start playing a phrase now, it will always send the specified control message (value 64 on controller 100, Midi channel 1) from the PA. Info output.

Now call a Modifier (Menu Modules> Modifier) in the Room view, put it right under the Player and connect the output "PA. Info> „ to the input "> Signal" of the Modifier. Then draw a connection from the Modifier’s output "All Midi>" to "> Control " (that is an input, so it is located on the left) of the Player.
Your room should now look something like this:



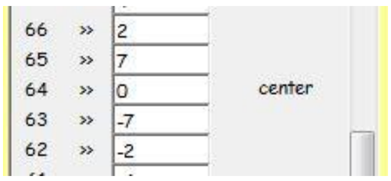
Open the Modifier’s editor and look for the words "Velocity / Contr. Value (0 ... 127) "**on the right hand side** (this line also appears on the left, but that is not interesting for us at this point).

Now click on the arrow to the right of it and find the line “rd. offst” (random offset). To the right of "Range", enter +2 and left of it -2. The goal here is that each new phrase is transposed by Lines randomly between two fifths upwards and two downward. So there are 5 possible values: -2, -1, 0, 1 and 2. Now close the editor of the Modifier.

The Modifier always gets the same value from Lines, namely 64. This simply means "Now a new phrase begins". Now the Modifier produces offsets from that value of 64. It is configured in such a way that it produces values from 62 to 66 (64+/- 0, +/- 1 or +/- 2) randomly and sends the result back to Lines.

Now the Player still has to learn to interpret these values 62 to 66 correctly, that is as a transposition of -2 to +2 fifths. For this, open the Player editor again.
Locate "Transpose (semitones)", click the *struck-through* R right next to it and you will see the corresponding Table, the conversion Table for Transpose in Track 1 of the Player.

Firstly, empty the Table by clicking Clear Table.
You know that you are only interested in the input values 62-66 which come from the Modifier. For these, you must enter values into the Table. Now you know already that you can enter the value 0 - "no transposition" for 64.
You also know that a fifth consists of 7 semitones and that for a transposition of 2 fifths you would have to enter + or - 14 accordingly. Since you want to avoid transpositions of more than one octave (12 semitones), you put +14 down one octave (14 - 12 = 2) and -14 up (-14 +12 = -2). So in the Table you need for 62 "-2" for 63 "-7", 64 gets "0", 65, "+7" and 66 is converted to "+2". Now you fill the remaining (never used) numbers of the Table with linear interpolation, just to have them defined and get this picture:



66	»	2	
65	»	7	
64	»	0	center
63	»	-7	
62	»	-2	

Finally, Lines has to know which controller number is being sent as PA.Info. For that you set the value next to the words “Midi controller”, on the left of the RTC editor, to 100.
Lines now plays almost every phrase in a different key and these keys are in a close fifth relationship.

The Lines Transpose Modifier can be found in many of my example-Rooms.

Now you can follow the cooperation of Players and Modifiers in the Player’s editor, when it plays. Via tooltips at the outputs and inputs of the
Modifier you can find out about the values of the last control messages, when you pass over it with the mouse. The tooltip for "> *Control* ", the RTC input of the Player tells you which controllers the Player is listening to and which parameters they control, as well as a tooltip at the output "PA.Info>" tells you which ones it sends.
These features are very useful when you are debugging complex Rooms.

Use Best Sample
If Lines runs in a track that plays audio, this track uses the sounds (audio, not Midi) which you played moments ago.
The audio data are available as samples in memory and Lines can play them note by note exactly in the order you recorded them. As that audio has been memorized as single notes, Lines can search for the most suitable note for this musical context, instead of using the original note as you have played it. And sound more natural this way.
An example of the usefulness of this parameter:

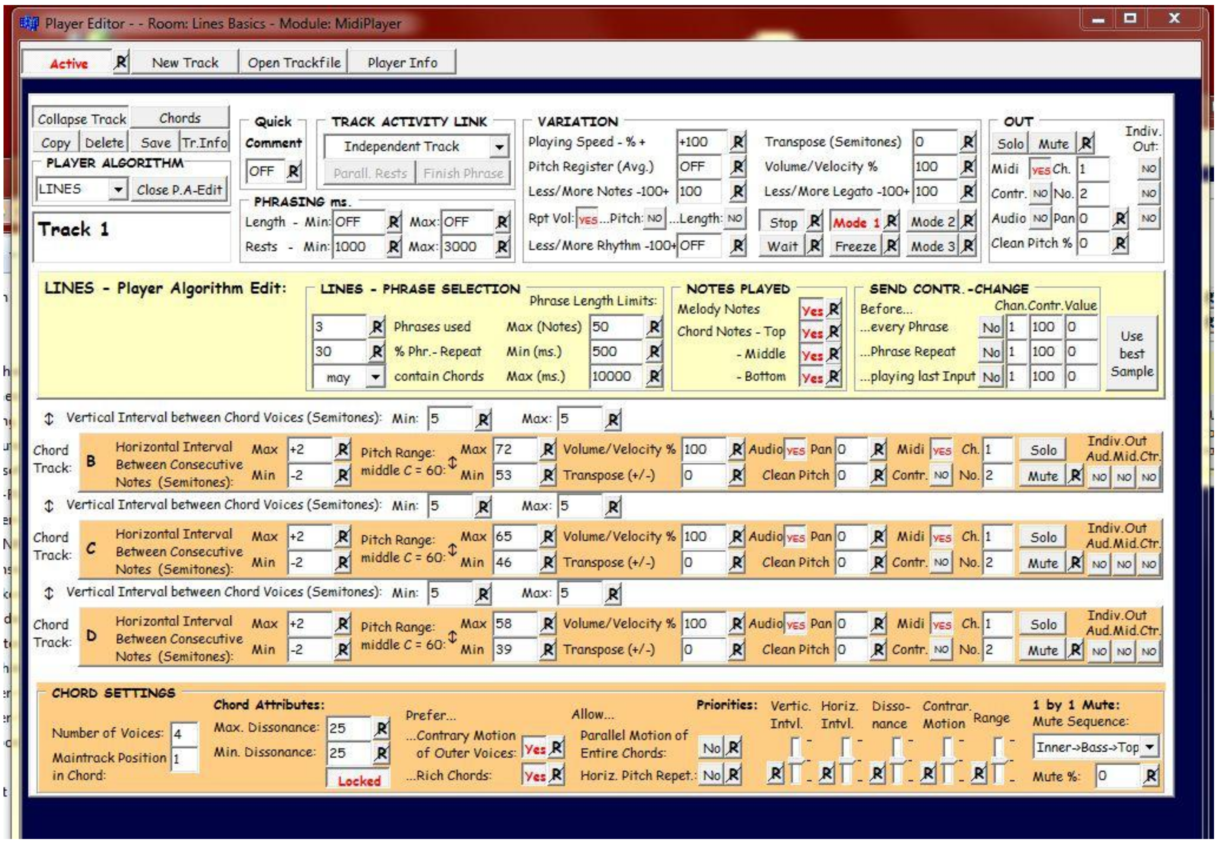
For harmonic or other reasons, you need an upward transposition of 7 semitones - a fifth - and set the "variation" Transpose to +7. In the setting "Use original sample" every note of the phrase will transposed upward by the Player by 7 semitones, thus often creating a Mickey Mouse effect. This effect is due to the fact that T² transposes formants together with the rest of the audio. But in nature formants remain unchanged while pitches of an instrument or a voice can of course vary. Formants are responsible for the timbre of a sound.

You have perhaps already played a note that was 7 semitones higher and whose length and volume also fits quite good to the current phrase. This note does not have to be transposed up by a fifth though, because it already had the desired pitch. If **Use Best Sample** is selected, the Player can find and use that note, even though it was not part of the currently played phrase. The effect: No Mickey Mouse anymore. This function also searches for length compatible samples so when speed changes, the sample length does not have to be corrected too heavily, which also sometimes leads to an unnatural sound.

With “**Use Original Sample if Possible”** transposition will take place, however as will possible necessary changes in length and volume of the notes. On the other hand Lines keeps the original context of the notes, so sometimes phrases can sound more natural with this setting. The best setting depends on the particular requirements of the current Room.

Formant (Mickey Mouse) effects frequently occur with transpositions of more than three semitones, sometimes even with less than that.

Player Chords


Player Editor - - Room: Lines Basics - Module: MidiPlayer

PLAYER ALGORITHM: LINES
Track 1
Quick Comment: OFF
TRACK ACTIVITY LINK: Independent Track
PHRASING ms.
Length - Min: OFF Max: OFF
Rests - Min: 1000 Max: 3000

VARIATION
Playing Speed - % +: +100
Pitch Register (Avg.): OFF
Less/More Notes -100+: 100
Rpt Vol: YES ...Pitch: NO ...Length: NO
Less/More Rhythm -100+: OFF
Transpose (Semitones): 0
Volume/Velocity %: 100
Less/More Legato -100+: 100

OUT
Midi: YES Ch. 1
Contr.: NO No. 2
Audio: NO Pan 0
Clean Pitch %: 0
Indiv. Out: NO NO NO

LINES - Player Algorithm Edit:

LINES - PHRASE SELECTION
Phrases used: 3
% Phr.- Repeat: 30
may contain Chords
Phrase Length Limits:
Max (Notes): 50
Min (ms.): 500
Max (ms.): 10000

NOTES PLAYED
Melody Notes: Yes
Chord Notes - Top: Yes
- Middle: Yes
- Bottom: Yes

SEND CONTR.-CHANGE

Before...		Chan.	Contr.	Value
...every Phrase	No	1	100	0
...Phrase Repeat	No	1	100	0
...playing last Input	No	1	100	0

Chord Track	Vertical Interval between Chord Voices (Semitones): Min	Max	Horizontal Interval Between Consecutive Notes (Semitones): Max	Min	Pitch Range: middle C = 60: Max	Min	Volume/Velocity %	Transpose (+/-)	Audio	Pan	Clean Pitch	Midi	Ch.	Contr.	No.	Indiv.Out Aud.Mid.Ctr.
B	5	5	+2	-2	72	53	100	0	YES	0	0	YES	1	NO	2	NO NO NO
C	5	5	+2	-2	65	46	100	0	YES	0	0	YES	1	NO	2	NO NO NO
D	5	5	+2	-2	58	39	100	0	YES	0	0	YES	1	NO	2	NO NO NO

CHORD SETTINGS
Number of Voices: 4
Maintrack Position in Chord: 1
Chord Attributes:
Max. Dissonance: 25
Min. Dissonance: 25
Locked
Prefer...
...Contrary Motion of Outer Voices: Yes
...Rich Chords: Yes
Allow...
Parallel Motion of Entire Chords: No
Horiz. Pitch Repet.: No
Priorities: Vertic. Intvl. / Horiz. Intvl. / Disso-nance / Contrar. Motion / Range
1 by 1 Mute:
Mute Sequence: Inner->Bass->Top
Mute %: 0

You will often play monophonic melodies with T² which then get stacked into a polyphonic texture by an algorithm such as Lines, using multiple tracks, possibly transposed and/or otherwise altered. If you are a horn Player (like me) you will work like that a lot.

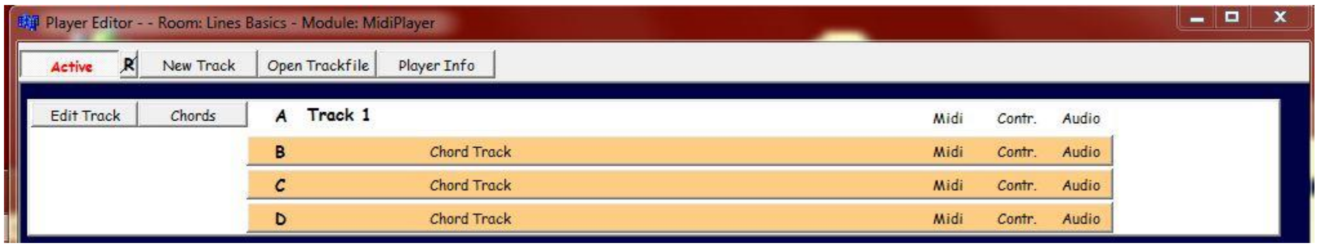
But sometimes you may not want a track to exclusively play melodic lines. Player Chords can wrap melodic material in chords.
The chords produced with this function rhythmically follow the melodic line this track is currently playing. You define a number of additional voices that T² adds to the original line. The new voices can be above or below.
This happens in real-time. T² tests many different chords, if and how much they correspond to your respective specifications. The choice then falls on the chord with the smallest "error". Here you can determine (and control remotely via RTC, if required) certain features of the chords generated in terms of dissonance, pitch range, intervals between the individual voices and horizontal intervals between notes, whether contrary motion between bass and treble is desired and whether pitch duplications are allowed in different octaves between voices. The various requirements that you set for chord voices and complete chords often imply contradictions. This is virtually inevitable. The priorities, according to which Player Chords finds and compromises between these "contradictory" demands, can also be determined by you.

Moreover, it is also possible to separately decide on Midi or audio output for each chordal voice or define individual outputs.
Play a few short monophonic phrases in BasisPlayer.room to make Lines begin to play.

To activate the Chord system please click on the button **Chords**. Now you can see that below the Player Algorithm editor of Lines the white track area has enlarged and new, light brown tracks appeared. They are the three chord tracks you have added, marked with the letters B, C and D. For track "A" you will look in vain, because that is the original Player-track, based on which the chords are produced.

Finally, at the bottom there is the **Chord Settings** box.
Here general chord properties are defined that are not attributable to individual chord Tracks.

A clearer, but not editable track display, you will see, as always, when you click on **Collapse Track**. Via
Edit Track (the same button) and then **Chord Settings** you get back to the editor, which we will use now.



A	Track 1	Midi	Contr.	Audio
B	Chord Track	Midi	Contr.	Audio
C	Chord Track	Midi	Contr.	Audio
D	Chord Track	Midi	Contr.	Audio

Chord Settings

Here general chord properties are defined that are not attributable to individual chord Tracks.
From left to right:

Number of Voices
This parameter determines of how many voices the chord consists, including the original track (the track on a white background).
Main Track Position in Chord determines whether, for example, in four-part chords the original track should be the soprano (1) alto (2), tenor
(3) or bass (4).
Which degree of dissonance should the chords have? Here, a range with minimum and maximum values or a fixed target value can be defined. For a fixed value the
"Locked" button must be set. The values of this parameter range from 0 (almost only octaves, unison, fifths and fourths - little dissonance) to 127. Here you find only chords with strong dissonances like major sevenths and minor seconds.
With a value of 15 the first thirds appear, still with many unison notes, at 25 you only hear triads, perhaps with the first occasional tritone in it.
Up to 40 the triads get richer, unison notes and octave doublings disappear quickly. At 50 the first major seconds, ninths and minor sevenths come in and at 70 you can hear some minor seconds and major sevenths. Above 70 fewer consonant intervals like fifths and thirds are played. Here more and more minor seconds / ninths and major sevenths start to dominate the chords.

Prefer...

... Contrary Motion of Outer Voices

For centuries, it is considered particularly elegant and well sounding, if the top and the bottom part of a harmonic progression move independently, possibly even in opposite directions. This contrary movement can be selected here.

Of course, you can also select the opposite, that is, the parallel movement of the outer parts, which frequently is found in jazz or pop music.

... Rich Chords

In the dissonance range below 50, where it is often about triads, sometimes the question arises whether a chord that contains only the root and third, but not the fifth is more suitable for the required dissonance and other needs than a complete triad. In this case already existing notes in the same or different octaves would be doubled, so that the chord contains as many notes as indicated in the parameter "Number of Voices".

If this preference is set however, the Player is more likely to go for the richer sound in which fewer notes are doubled.

Allow...

...Parallel Motion of Entire Chords

The rules by which chords are generated sometimes result in situations in which the best solution for T^2 seems to be the parallel transposition of complete chords. This obviously leads to a simultaneous parallel movement of all voices, sometimes very attractive, sometimes not. You decide.

...Horizontal Pitch Repetitions

Especially middle voices (alto and tenor), sometimes get as the next note a repetition of the last pitch. Thus, the progression matches best with your specifications but on the other hand, the lyricism and autonomy of the concerned chord voice may be suffering. You can allow or disallow these repeated notes here.

Priorities

A key parameter of the chord module:

Please imagine the following situation: The bass has just played a low Eb - for the specified pitch range in the bass track (track "D") this corresponds to the lowest allowable note (Midi pitch 39). The top voice, which is the original voice played by the track is just in an upward motion, let's say it has a treble C and its next note would be D. You have opted for counter-movement in the Chord Settings and as the upper voice indeed is the original voice and cannot be influenced, the bass would have to go down to provide contrary motion.

This contradiction demands a decision, a compromise: either the bass leaves the allowable pitch range (plays a lower note than Eb 39) or the preference for contrary motion is violated.

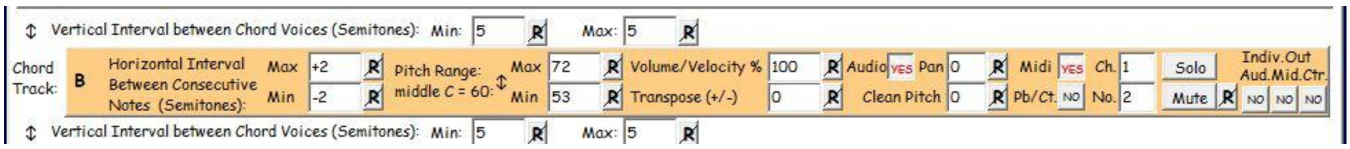
Such compromises and decisions are possible with the priority-faders. Set the right fader ("Range") all the way up so the range preference ("bass pitches not below 39") will be complied with. If the fader for contrary Motion is pulled up, the range fader will return to the middle, because in a compromise both parties never can take their positions to 100%. The counter-movement will take place, but the bass violates its given pitch range.

Discrepancies like this are inevitable. Here you can determine which parameters get the upper hand and which are to be treated more flexibly. With the help of the RTC system, you can also change priorities in certain musical contexts without having to use the mouse.

1 by 1 Mute enables you to gradually thin out chords voice by voice. 0% means that all chord voices are played and 100% means silence. In what order voices are muted is determined in the selection box below: Inner> Bass> Top means (for four-part chords) that first the two middle voices, then the bass and finally at 100% the top voice is muted.

Chord Tracks

Consider now the upper **Chord Track (**B):
Here, from left to right, you can make the following settings (or configure their real-time control with RTC)



Vertical Interval between Chord Voices (Semitones): Min: 5 Max: 5

Chord Track: B
Horizontal Interval Between Consecutive Notes (Semitones): Max +2 Min -2
Pitch Range: middle C = 60: Max 72 Min 53
Volume/Velocity % 100
Transpose (+/-) 0
Audio YES Pan 0
Clean Pitch 0
Midi YES Pb/Ct. NO
Ch. 1 No. 2
Solo Mute
Indiv.Out Aud.Mid.Ctr. NO NO NO

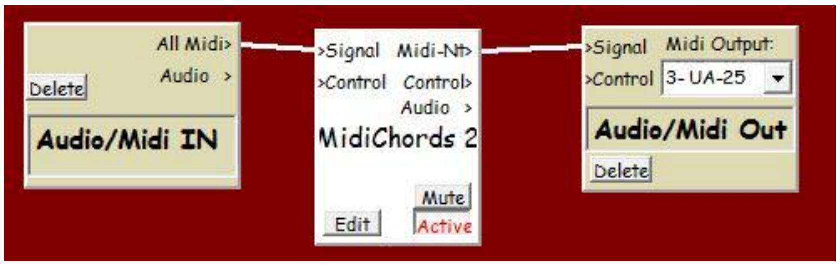
Vertical Interval between Chord Voices (Semitones): Min: 5 Max: 5

Horizontal Interval Between Consecutive Notes
determines how big the pitch interval between **two consecutive notes in THIS chord voice** may be (minimum/maximum). This is a voice leading rule with which you can prevent, for example, a voice to make jumps bigger than major seconds (more than 2 semitones up or down) from note to note. In the example, major seconds up (+2) and down (-2), minor seconds (+ / -1) and repeated notes (0) are allowed.

Pitch Range
is defined for every voice separately. The representation is in Midi-numbering, middle C has a pitch of 60.
The following parameters such as **Volume, Transpose, etc.** you know from the track editor. Transpose here refers to **this** chord voice. Since the other chord voices are not affected, changes here lead to changed situations within the chords, for example, in the vertical intervals, the range and often in the dissonance. However, if you change the transpose setting in the white main track, the entire chord will be transposed. The aforementioned chord properties then stay unchanged.

Vertical Interval Between Chord Voices
Between the chord tracks, you will find the settings for intervals between voices.
Vertical here means that this is about pitch distances between simultaneously sounding notes **of different** chord voices (intervals are called vertical, because in western notation simultaneously sounding notes appear one below the other in a vertical column).
For example, if the lowest voice, the "bass" plays a low C (Midi-pitch number 48) and you set 5 as the smallest and largest distance between voices - a forth - the next voice of the chord, the "tenor" should have pitch 48 + 5 which is F (Midi: 53).
But maybe this F does not fit for harmonic or other reasons here. In this case, T² searches for a nearby pitch that meets the harmonic and other requirements better. For more on the prioritized search for "compromises" between conflicting requirements, refer to the preceding section on Chord Settings.

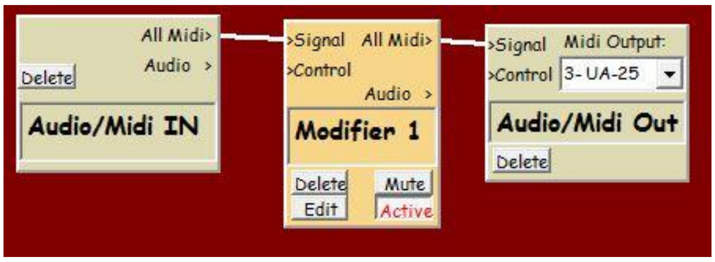
If you like to try out "Chords" with real-time material played live from your keyboard, simply call



an In- and an Out-module and put "Midi Chords "from the Modules menu in between. The editor has a somewhat outdated user interface, but controls the same parameters. Now you can surround your live melodies with chords and play around with the chord parameters. The module “Midi Chords” only works with Midi signals. In the Player, however, you can use the chords with audio samples. Depending on your taste (and your intonation on the instrument) the chords sometimes sound better with a bit of Auto Tune ;-)

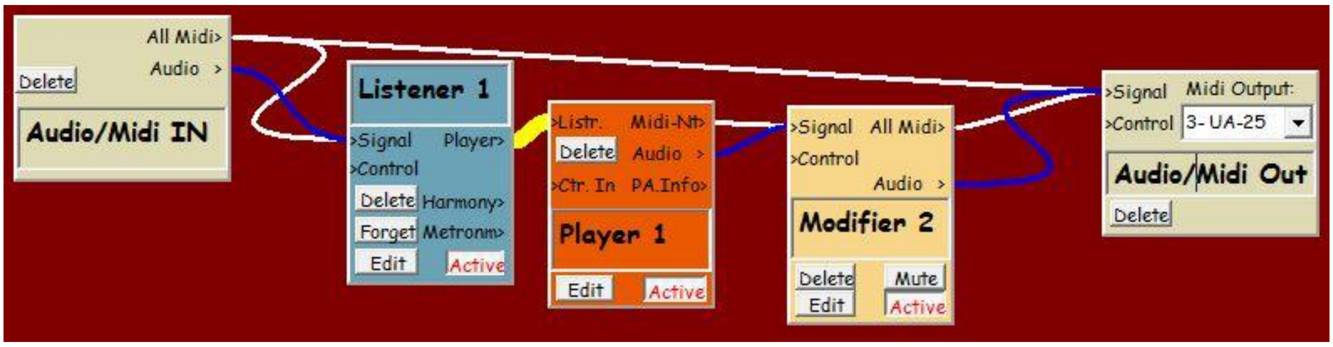
Note that “Chords” here has no implications in terms of “harmony”. Harmonic knowledge will be integrated by the new module “Harmony”, which will be hopefully available by the end of 2015. See the chapter “Harmony” for this.

Modifier



The Modifier serves to change particular note- or control-parameters such as pitch, note length and interval of an audio or Midi signal.
It can be located between an In- and an Out module to work in real time with Midi signals, or to modify signals coming from or going to any other module.
As it is convenient to test the functionality of the Modifier with the real time connection, I suggest configuring the Room like this:
Save the Room as ModifierThru.room.
If you would like to modify an audio signal, you have to set up the Modifier so that it functions to change an audio signal, which comes from a Player and **not** from Audio-In.
As the Player can send Midi or audio signals, you can change between audio and Midi signals in this configuration with these same Modifier settings. Audio however, does not work with real-time signals as they come "live" via Audio In.

To connect a Modifier for audio operation, start from the well-known BasisPlayer.room and open a Modifier in the modules menu. Now replace the connection from Midi-Nt of the Player to "Signal" input of the Audio/Midi Out module by one that goes from the Player's audio output to the Signal input of the Modifier and from the Modifier's audio output to Signal of the Audio/Midi Out-module.
Cancel connections by redrawing them with the Ctrl key pressed.
Of course, you can also leave the appropriate Midi connections as they were.
Save as BasisPlayerModifier.room, which could look something like this image.



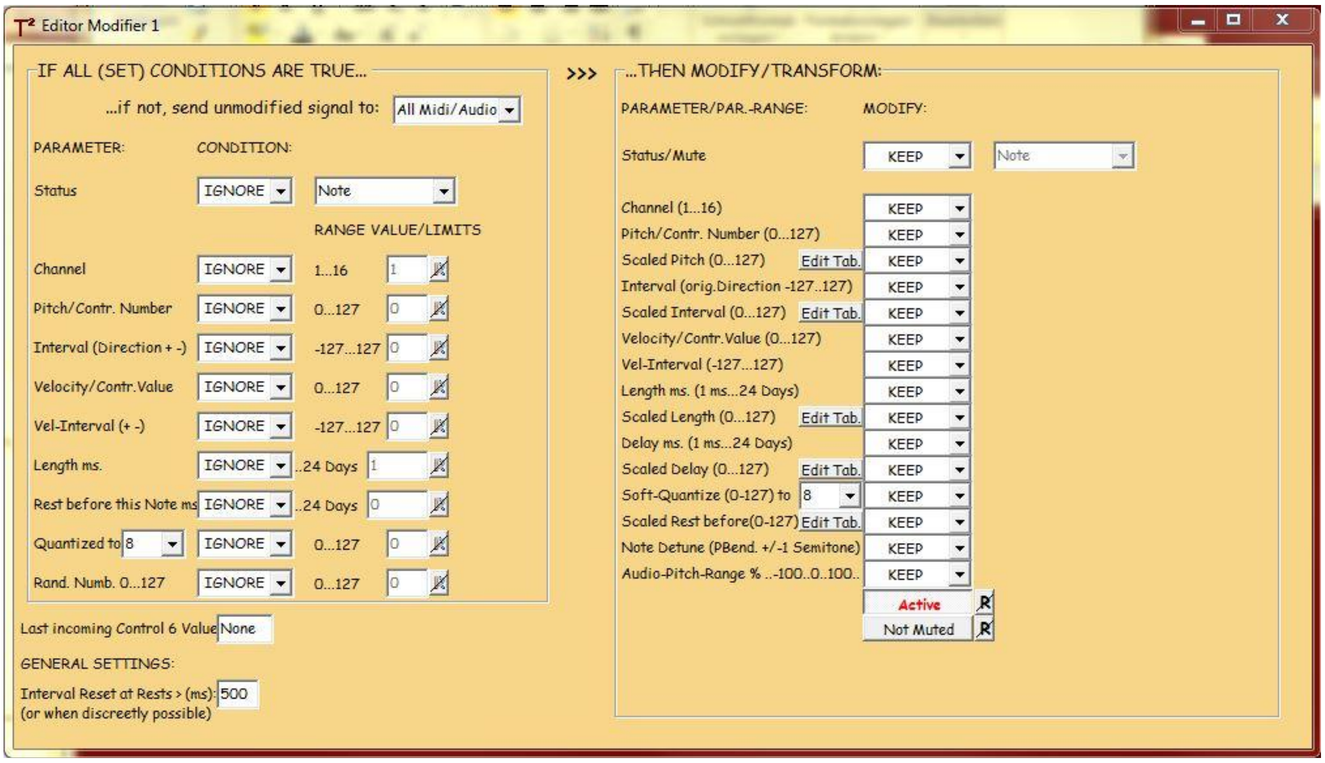
This Room can also modify note lengths.
In contrast, this does not apply to "ModifierThru.room". Because of the real-time ("through") situation, the notes are not yet known to the end, when they are to be changed by the Modifier.

Often you will use multiple Modifiers in a Room, each responsible for different tasks.
You can change not only notes but also streams of controllers. In the Midi-language these messages look like notes, only that they carry the status of Control Change (176) instead of the note status (144).
Where the pitch is encoded in the note messages, controllers have their controller number (e.g. a modulation wheel is represented by controller 1, a sustain pedal is a controller 64 etc.)

and instead of the note’s volume (in Midi-Lingo called Velocity) the controller value is transmitted. The „value” (0 ...127) carries the information of a control message.

The concept of the Modifier is based on Werner Kracht’s "Logical Edit" (as found in Cubase), but in its possibilities it goes beyond that.

As elsewhere in the program, open the editor of the Modifier by clicking on EDIT in the Modifier in the Room display.


T² Editor Modifier 1

IF ALL (SET) CONDITIONS ARE TRUE...

...if not, send unmodified signal to: All Midi/Audio

PARAMETER:	CONDITION:	RANGE VALUE/LIMITS	
Status	IGNORE	Note	
Channel	IGNORE	1...16	1
Pitch/Contr. Number	IGNORE	0...127	0
Interval (Direction + -)	IGNORE	-127...127	0
Velocity/Contr.Value	IGNORE	0...127	0
Vel-Interval (+ -)	IGNORE	-127...127	0
Length ms.	IGNORE	..24 Days	1
Rest before this Note ms	IGNORE	..24 Days	0
Quantized to 8	IGNORE	0...127	0
Rand. Numb. 0...127	IGNORE	0...127	0

Last incoming Control 6 Value None

GENERAL SETTINGS:

Interval Reset at Rests > (ms): 500
(or when discreetly possible)

>>>

...THEN MODIFY/TRANSFORM:

PARAMETER/PAR.-RANGE:		MODIFY:	
Status/Mute		KEEP	Note
Channel (1...16)		KEEP	
Pitch/Contr. Number (0...127)		KEEP	
Scaled Pitch (0...127)	Edit Tab.	KEEP	
Interval (orig.Direction -127..127)		KEEP	
Scaled Interval (0...127)	Edit Tab.	KEEP	
Velocity/Contr.Value (0...127)		KEEP	
Vel-Interval (-127...127)		KEEP	
Length ms. (1 ms...24 Days)		KEEP	
Scaled Length (0...127)	Edit Tab.	KEEP	
Delay ms. (1 ms...24 Days)		KEEP	
Scaled Delay (0...127)	Edit Tab.	KEEP	
Soft-Quantize (0-127) to	8	KEEP	
Scaled Rest before(0-127)	Edit Tab.	KEEP	
Note Detune (PBend. +/-1 Semitone)		KEEP	
Audio-Pitch-Range % ..-100..0..100..		KEEP	

Active R
Not Muted R

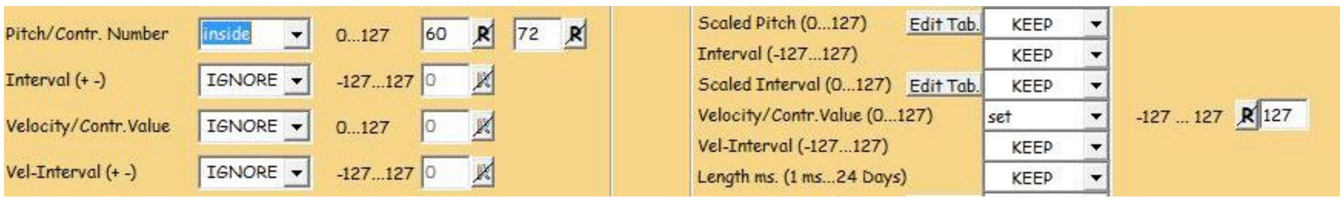
The editor is split into a left and a right side:

 On the left, you have the option to define the conditions under which the right side will make changes in the note parameters. If no conditions are formed (that is, if “IGNORE” is set beside all condition parameters), the right side will always automatically function according to the parameters defined there, unless the Modifier is inactive or muted (silenced). If more conditions than one are set, they must all be fulfilled (programmers call that the “logical AND”).

And now for the parameters on the left side, the…

Modifier Conditions

Define the kind of conditions you want with the dialogue box that now shows “Ignore.”
The limits will be defined to the right of this.
For example, you can specify: “Only get active when the pitch of the following note is between Midi note 60 and 72, or middle C and the C above that. Then make the note extremely loud.” To this end you will want to choose “inside” instead of “ignore” in the line next to “Pitch/Contr. Number” and define the limits of this pitch area in the number fields that appear to the right. These values include the two limiting values (60, 72) for both “inside” and “outside” respectively, that is to say, the conditions that apply in the above example are also fulfilled when the value is 60 or 72. To make the notes very loud once the conditions are met, you need to search for the line “Velocity/ Contr. Value” in the Modify area to the right, choose “set” instead of “keep” here, and enter 127 as the value. The editor would then look like this:



Pitch/Contr. Number	inside	0...127	60	72
Interval (+ -)	IGNORE	-127...127	0	
Velocity/Contr.Value	IGNORE	0...127	0	
Vel-Interval (+ -)	IGNORE	-127...127	0	

Scaled Pitch (0...127)	Edit Tab.	KEEP		
Interval (-127...127)		KEEP		
Scaled Interval (0...127)	Edit Tab.	KEEP		
Velocity/Contr.Value (0...127)		set	-127 ... 127	127
Vel-Interval (-127...127)		KEEP		
Length ms. (1 ms...24 Days)		KEEP		

Parameters that can form conditions are

Status: what kind of message are we looking for? Possible are notes, control changes, pitch bend etc.

Channel means the Midi channel.

Velocity always stands for volume level in the program. Its value can lie between 1 and 127.

Vel-Interval means that the difference between the velocity of the last note and the current one must be past (or be inside) such-and-such a boundary (e.g. highest or lowest value) or the Modifier will not work.

Interval is also about differences, but with respect to the pitch of two consecutive notes.
In the condition area, the interval-directions are taken into account. So if you want to have the Modifier search for descending major thirds, you must enter "- 4" (and not just "4").

Length
Note length in milliseconds. As I said, this is not for real time operation, but only for notes that come from the Player.

Rest before this note: Notes that come after a rest often have a greater importance. With this function, you can refer to these notes separately.

Quantized to...
When a Metronome connected to the control input of the Modifier defines a tempo, you can configure how near in time a note must be situated to a quarter-, eighth- or sixteenth-note of the given tempo for the Modifier to change it. On the far left you will define a quantization

grid, then the conditions, and finally the value. “Value” here means that 127 is (time-wise) an “exact hit” on the grid, and 0 represents a note exactly halfway between two defined points on the grid.
So the proximity of a note to a quantization point is converted into a value – which in turn is evaluated by the set condition.

Random Number:
With every note, a value between 0 and 127 will be chosen at random, and when the resulting value satisfies a defined condition, the right side moves into action. You can specify the statistical frequency of a modification without knowing exactly when it will occur.

RTC for Conditions

To the right of the filter parameters you find the well known struck-through "R"s. Here you can configure real-time control functions.
It works exactly like in the Player (refer to the RTC section in the Player Chapter), only that here you are dealing with **conditions** that can change dynamically via RTC.
It is possible, for example, to set up a function that ensures if you play rather a lot, **the chance increases** that notes are muted by the Modifier so that T² in turn plays less under these circumstances.

For this you would use the Listener parameter "Note Activity", which measures the amount of your playing, and connect it with the Modifier’s Control In (green connection) in the Room display. In the editor of the Modifier, next to "*Rand. Number*„ choose "*smaller*" and then click the R button. Finally, in the top right corner of (Status / Mute) you select "set" and on the right next to that "Mute".

Now, for every note that reaches the Modifier a number between 0 and 127 is chosen randomly. If that value is smaller than the condition value (which in turn is controlled by the Listener) this note will be muted. Since the chance that the random number is below the threshold rises with increasing activity of yours, more notes will be muted the more you play. You can watch RTC control the limit value in the Modifier’s editor.
The Listener’s operation will be explained later, this paragraph is simply meant to be a glimpse at the possibilities of RTC.

If the mouse hovers over an R button, you can see the RTC parameters (Midi channel and controller number) as a tooltip.
To find out which messages have passed an input or output of modules, simply move the mouse over the in-/output to see the tooltip.

So much for the “Conditions” side of the Modifier editor.

Above the conditions on the left side of the dialog, you will find the words
"... If not, send unmodified signal-to:" and then a selection menu. This provides two outputs, which can be found in the Room display of the Modifier.

Here it is possible to configure more complex modifier setups with branches. If a condition for an event (note / Midi controller) is not satisfied, the event is usually passed unchanged to the next module. Just as if the Modifier is not "active", but switched to "thru".

You may prefer though that the event in this case (condition not met) will **not** be forwarded at all, or that it will be passed to a different module. In the first case, you would select the Aux output here, but connect nothing to it. In the second case, connect Aux Signal to another module. Aux works for all kinds of data (Midi-, audio-notes or controllers)
Now it is possible to use Modifier conditions for defining the path a signal takes in a room.

Below the conditions there are two more general parameters, which are logically outside the framework of conditions and modifications, and therefore apply for the whole editor:

Last Incoming Control 6 Value
Sometimes it is useful to cache a value in the Modifier. For this purpose, the last value of control 6 - messages that reached the Modifier appears in the box to the right. The C6- values must reach the Modifier over a green connection
(i.e. the control input).

The Midi channel is irrelevant.
The Modifier uses this function to replace certain parameters of messages, such as the pitch or velocity of notes with the value stored here before passing it to the next module.
See the section on "Use Instead“ below for this modification method.

Interval Reset at Rests > (ms)
(or when discreetly possible)
This parameter defines that notes which follow a pause longer than the given value (in milliseconds), won’t be heard by the Modifier in context with the preceding note (and therefore don’t form an interval).
With a setting of 500 milliseconds, an interval between two notes would be changed only if they are separated from each other by less than a half second. But if the rest between two notes is longer, the interval remains unchanged or (if set as a condition) will not be taken into account. Often you may want an interval adjustment between two notes only when they have a close relationship to each other within a certain amount of time, e.g. not at the beginning of a new phrase or between unconnected notes.
This parameter is valid for all interval settings of this Modifier (pitch and velocity) - on the filter and the modify side.

... **(or when discreetly possible):** Since certain interval combinations sometimes produce very high or low notes, there is a mechanism which resets intervals at inconspicuous spots regardless of the "**Interval Reset at Rest"** setting.

On to the right half of the Modifier editing page:

Modify/Transform

Look for “Pitch/Contr. Number...” on the right side and on that line, click on the arrow next to “Keep,” in order to get an overview of the functionality.

The Options box offers various…

Modification Procedures

with which these parameters can be changed. The numerical values to be edited, Tables or other settings appear only after a modification method was selected.

When you select **“Set”,** you’ll find to the side a numeral input field in which you can enter a value to which the parameter will be changed.

Easy **mathematical operations** follow and with multiplication and division, floating point numerals can also be entered.

Since minus operations are possible, this can lead to negative controller values. This is no problem between modules of T², although negatives do not correspond to the Midi standard. Before sending them to the outside world, don’t forget to convert them back to positive values.

Sometimes this is necessary if, for example, a controller value oscillates around 64 and you want to reduce deviation from this value by half. For this you first subtract 64 to make the value oscillate around 0, then use another Modifier to divide the value by 2, and finally add 64 in a third Modifier - done. A similar thing happens for instance in the example Room TwoDuosForHorn.room, for which there is also a music example, on the far right at the bottom of that Room.

In **“Random Set”,** T² randomly chooses the value from a value range defined by you, while

“Random Offset” randomly changes value within the given boundaries.

“Reverse” inverts values around a center value chosen by you. The incoming value of 56 will result in 64 with a “Center” of 60 (60+4 instead of 60-4), and for input 70, the result with the same “Center” would be 50.

“Table-Transformation”: Here you have the possibility to define a Table, if the desired conversions cannot be specified with a simple math operation. User-defined values can be assigned to given input values between 0 and 127, which then replace the original value and emerge as the output of the Modifier. If necessary, you can also put the arbitrary values to the left side of the Table. The button “Switch Mode Conversion” in the Table editor enables therefore the output of negative and values outside the range of 0 to 127.

“Use Instead” allows you to select another parameter of a note through which the parameter in this line will be replaced. For example, you could replace the pitch of a note (from 0 to 127) with its volume (velocity, the same value range). This would mean that the louder a note is before the Modifier, the higher it will be at its output. This function can also use the cached

“Last Incoming Control 6 Value**”.**

“To Control Out” sends the note unchanged to the Modifier’s output and simultaneously sends a control message (Control-Nr. 6), which contains the value of this parameter, on the same Midi channel.

The Individual Modifiable Parameters:

Status/Mute offers the possibility of transforming Midi data into another kind of data or deleting it through Set->Mute (e.g. with defined conditions). With the exception of status and channel, all other parameters work similarly with Midi and audio notes.

Channel, pitch and velocity are the main parameters for notes. Because the Modifier can change other Midi messages as well, pitch and velocity are also labelled as “Cont. Number” and “Cont. Value” respectively.

Interval and Velocity Interval compare the pitch/volume of the last note with the value of the current note, if both are situated in a sufficiently close proximity in time. The definition of “sufficiently close proximity in time” is set in “**Interval Reset at Rest**” in the lower left. Interval modifications respect the direction of the original interval. "Set 3 "thus results in a pitch a minor third upwards when at the input of the Modifier the interval was also upwardly directed and vice versa.

Length and Delay lengthen/shorten or delay notes, while…

Soft-Quantize pulls the notes onto a freely selectable quantization grid.
This is only possible if a Metronome is connected with the Modifier via a black wire and relays the tempo information to it.
“Set” 64 for example, would mean that every note will be placed exactly in the middle between its original time-related arrival and the next quantization step. “Set” 0 means it stays where it was and “Set” 127 signifies that it will be placed exactly on the quantization point. The next quantization point is defined as the time at which the next note of the left-specified value will occur (in the tempo given by the metronome). 8 means eighth notes here, bar is the “one” of the next bar and the points refer to dotted note values.

The initial Legato-situation of the notes is reconstructed after quantization. This way unwanted Staccatos or overlapping notes will not occur.

This is how a soft and (with the help of the remote control) even variable quantization works. You can produce “streams of notes” that behave like swarms of fish, moving in a coordinated fashion, yet not all rigidly jumping together to affix themselves to the same point at the same time.
Try SoftQuantize.room with the modulation wheel (“Info” is available).

The parameters that begin with
Scaled... first convert the incoming values using a Table into a scaled standard parameter, perform the given operation, and finally use the same Table to convert the value back again to the original measure of the respective parameter. Confused? Here’s an example:

Scaled Length translates millisecond values on a scale from 1 to 127. In this case it is logarithmic. The middle (scaled) value, 64, is equivalent to 1000 milliseconds; 96 stands for 2000 milliseconds, 112 stands for 3000 milliseconds and 127, the highest possible standard parameter value, is dedicated for note lengths from 10,000 milliseconds (10 seconds) and more. The values of this conversion are chosen to form a curve.
On the left of the word "KEEP" you can view and edit the Table.

In this case a logarithmic curve helps a lot:
If I want to prolong each note by a quarter of a second (add 250 ms. to the length), the difference will be very noticeable at short notes (100 ms. becomes 350 ms.), but almost imperceptible at notes of e.g. 2 seconds in length; numerically the same number of milliseconds has been added (from 2000 to 2250 ms.).

When I use the scaled parameters, I might initially define "lengthen every note by 20" without knowing what "20" is. But I know that 20 is about a sixth of the entire range of the standard parameter (0…127), so I can make a closer guesstimate of the change I would like.
Now the actual length of the note will be converted into the standard parameter; in the case of a 100 millisecond-long note, the standard parameter's value would be 9. I add 20 to that and get 29. The millisecond value belonging to 29 is 365 and with that, the result is very similar to that yielded when (as above), I just add 250 milliseconds.

Now I add 20 to a note which is 2000 milliseconds long:
2000 milliseconds, according to the Table, is equivalent to a value of 96. 96+20=116, and 116 is equivalent to 3465 milliseconds – a "**felt**" difference which appears to be more like the perceived difference between 100 milliseconds and 350 milliseconds but which actually adds 1500 to the note length, from 2 seconds to nearly 3.5 seconds. In this case, a multiplication with 1.75 would have done the job for the long note, but for the short one we would have needed a different factor.

Thus Tables also have the task, among others, of converting numerical values in such a way that they correspond to our perceptions. With intervals, this is of course totally different than with the length of notes, rests or pitch; for this reason there are various different Tables, which can, of course, be edited. Tables are saved and opened with their corresponding Rooms.

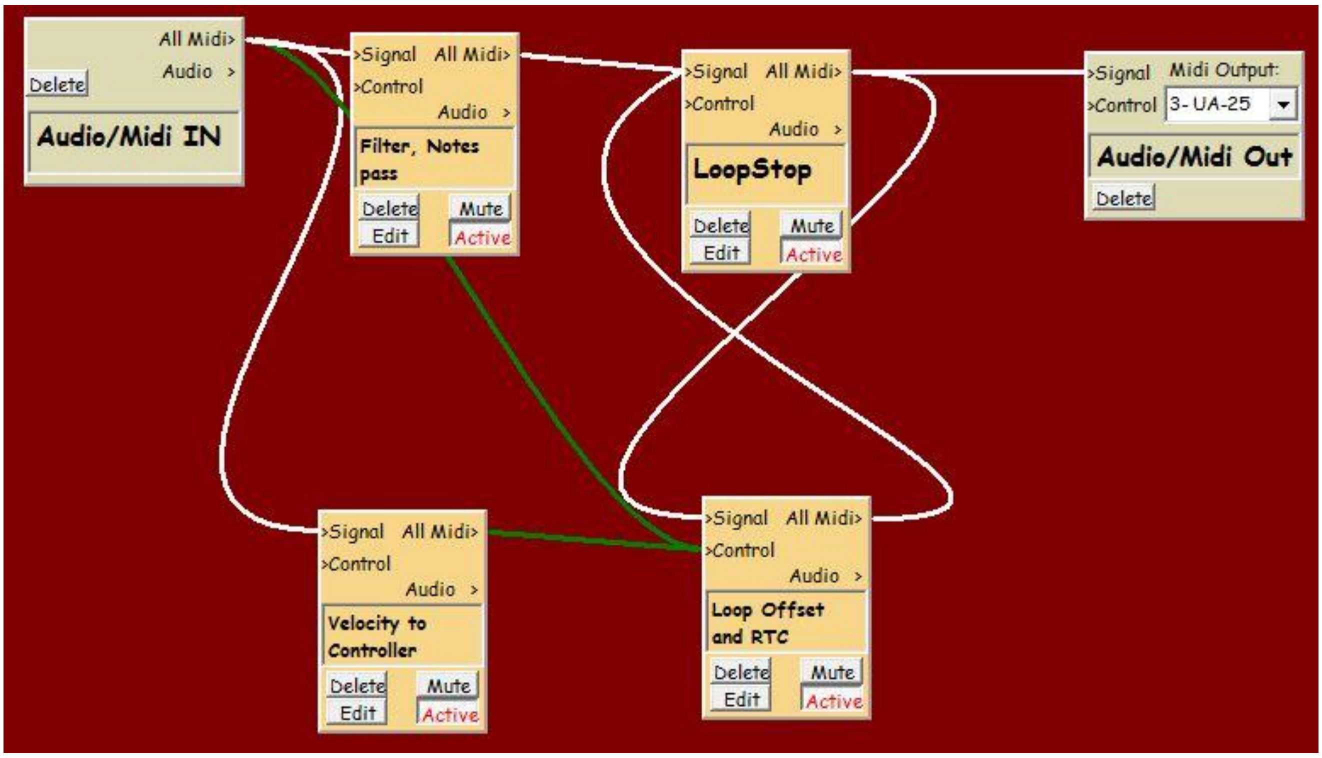
Note Detune: The Midi and audio notes can be tuned here. A value of +64 means a tuning up to the next-highest half step (for example G# instead of G). I have chosen +/-64 as the range (against the traditional cents standard), because it is functionally equivalent to the Midi pitch bend. The Modifier changes both Midi and audio pitches, so that also single notes from Midi tone generators can be detuned in a flexible way. Important for the scaling of the detuning of Midi sounds is simply that the value for the pitch bend range in the connected tone generator and that in "Extras menu> Global Audio Settings " match.

Audio Pitch Range: Every audio note, unlike synthesizer or piano notes, runs a one-time, i.e. unrepeatable course in regards to volume and pitch. T² can detect these tiny abnormalities down to a resolution of 1/100 of a second. With this data, which the Modifier gets with every audio note, it is in the position to balance out, stress or even invert the smallest pitch discrepancies. The percentage refers to the actual pitch discrepancy: 100% means "no change compared to the original note," –100% means a reverse discrepancy, 50% means half, and with 0%, you get notes without a pitch gradient, somewhat as if you were playing the piano. 0% here has the same effect as 100 % of Auto Tune in the Player.

This function also depends on the globally adjusted pitch bend range
(Extras menu> Global Audio Settings) and it works with Midi as well as with audio notes.

Here are some examples of Rooms with Modifiers. Behind the Rooms' "INFO" buttons you will find descriptions of the Rooms. They are located in the folder "Technical Demos".
MidiOnlyFeedback.room
ContrToNotes.room
TempoFromPitch.room
DelayTriadControl.room

For example it is possible to arrange two Modifiers in a feedback loop to produce streams of notes with them and then control them via Real-Time-Control in a very flexible way. An example for that you can find in the Room MidiOnlyFeedback.room. Do not forget to use the modulation wheel there. More details in the Room's "INFO".



(MidiOnlyFeedback.room)

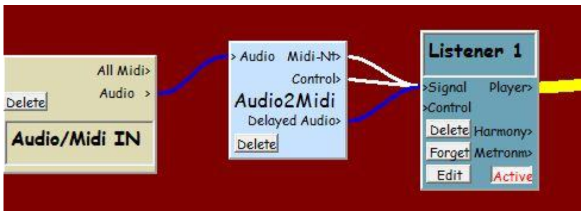
Audio To Midi (A2M)

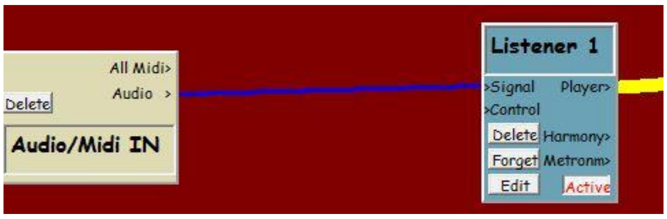
General information for dealing with audio signals in T²:
(Parts of this information can be found also in the "2 Interlude - The Player and Audio Signals". But since they are equally important for the understanding in both places, I think this redundancy can be forgiven.)

Audio signals and Midi signals are internally represented as single notes. Tango's Audio To Midi Converter (A2M) separates the individual notes from the audio stream, describes them with a resolution of 10 milliseconds in terms of volume and pitch and finally stores them separately in the Listener's memory, along with their descriptions.

The audio detection is looking for usable pitches**.** If you sing and use text, you will mainly hear the vowels in your singing, because only these contain recognizable pitches.

The Audio To Midi module can be called separately, but it is also integrated in the Listener and therefore does normally not appear in the Room. A (blue) audio connection from the audio output of the In-module to the Signal input of the Listener is sufficient. Thus, both the following setups have the same functionality:





The stand alone-A2M can still be useful if you want to use it outside the Listener simply as a "Pitch To Midi" device. It has 3 outputs, where the audio, Midi notes and the describing controller are put out separately. Via auxiliary programs such as LoopBe you can connect T² with other Midi programs running on the same computer. If the controllers or the audio signal are not needed, simply connect only the "Midi Nt" output.
Note that all output signals coming from A2M carry a delay of about one-third-second (see below).

Audio To Midi works only with monophonic signals like horns, voice or single note piano.

This system enables the Player to use previously heard audio material for its musical responses by reshaping it, combining notes to chords or having them changed by other modules. So T² can not only use pre-produced synth or sampled sounds, but uses your own sound to interact with you. Of course, the Player can also combine Midi and audio sounds.

Tango's A2M, as I said before only processes monophonic material, so overlapping notes, such as playing very legato on a piano or un-muted guitar strings as well as thick reverb or delays can disturb the pitch tracking. Ideal are dry horn and vocal signals. Even for the guitar, I would prefer a Midi solution.

Setting correct levels for audio material, that is as high as possible without clipping, improves pitch tracking. You can check the audio output of the in-module for that.

On the other hand, during your concerts you should hear the **output of T²** (the Player's output) as quietly as possible, so that you are still able to react to the program but avoid feedback of Tango's music blasting from the speakers into your microphone (that is into Tango's Listener). Feedback loops can provide a very interesting application of the program, but they should happen inside the program and not between microphone and speakers. (See below "Audio Gate")

Only the left of the two stereo channels gets analyzed.

The Outputs of the Audio To Midi Module (A2M)

A2M represents each audio note first with note-on and note-off Midi messages about its initial volume (velocity) and pitch (pitch).
Later (after the note-on and before the note off) the volume is further described with controller 2 (often called "Breath Control") and the changes in pitch with pitch bend.

The pitch and volume description happens at a resolution of 1/100 second and is stored in context with the samples in the Listener's memory.
These pitch and volume descriptions are put out by Audio To Midi via the

"MidiCtr" output.

"Midi-Nt" serves only for the note-on and note-off information and

"Delayed Audio" sends the analyzed audio signal with the same delay as the Midi data, so that audio and Midi occur synchronously.

The Listener gets the audio events about 0.37 seconds after you have played them. This value can be different on your system because of the potentially different audio latency of your hardware, but most of the delay has to do with the operation of my Audio to Midi module, therefore, it has technical reasons and cannot be switched off.

For T² this is early enough and it allows for good accuracy in tracking.

In A2M Output.room you can hear how the module works and experiment with it. For explanations, as usual, go to the "INFO" button in the Room display.
Because they are valid for Tango's entire audio system, the parameters for the Audio To Midi function are globally stored in the

Extras menu> Global Audio and Midi In Settings.

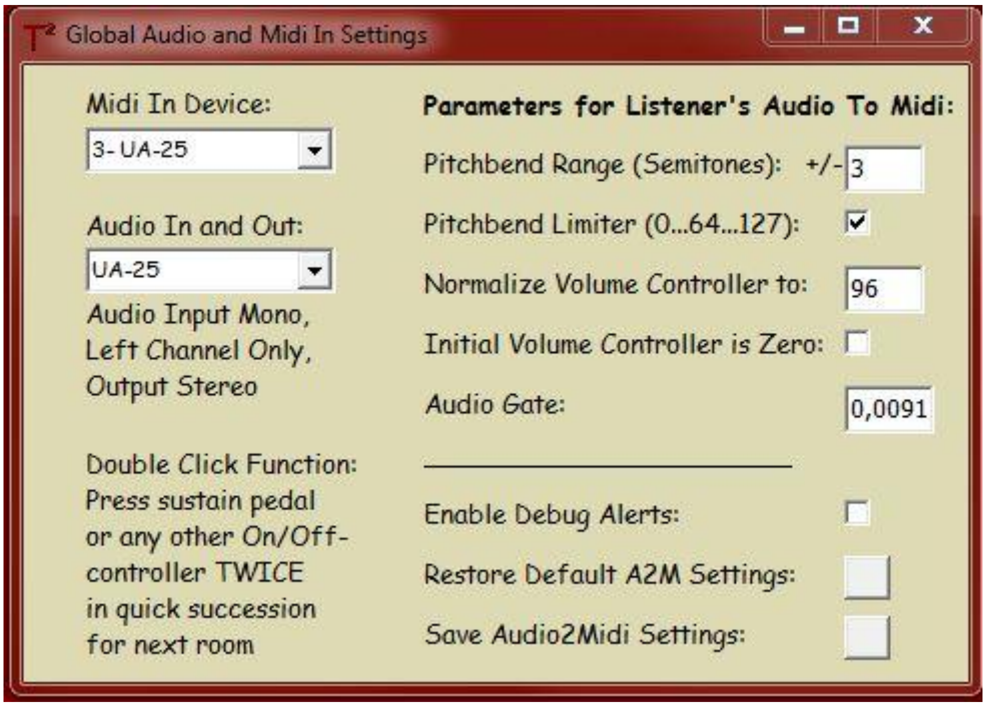
You can save your settings here. They are automatically loaded at startup.

Pitch Bend Range
The pitch bend range is preset to + / - 3 semitones by me. You can change that value.
All glissandi in the audio with a distance of no more than three semitones from the original pitch will therefore be described correctly in this setting. If you want to control a Midi- tone generator with these pitch bend data, you should have the same pitch bend range adjusted there so that the result sounds like the audio original.

Greater settings (e.g.12) may represent larger glissandi, but at the cost of pitch accuracy.

Smaller settings of course, are more accurate but cover only smaller bends.

The reason for this the fact that Midi pitch bend can take 128 different values (0 to 127 with a middle of 64) and only in the connected tone generator - or in Tango – one defines for how many semitones (+/-) the extreme values of 0 or 127 stand. Plus/minus 3 semitones has proven to be a good compromise.


T² Global Audio and Midi In Settings

Midi In Device: 3- UA-25

Audio In and Out: UA-25

Audio Input Mono, Left Channel Only, Output Stereo

Double Click Function: Press sustain pedal or any other On/Off-controller TWICE in quick succession for next room

Parameters for Listener's Audio To Midi:

Pitchbend Range (Semitones): +/- 3

Pitchbend Limiter (0...64...127): ✓

Normalize Volume Controller to: 96

Initial Volume Controller is Zero: ☐

Audio Gate: 0,0091

Enable Debug Alerts: ☐

Restore Default A2M Settings:

Save Audio2Midi Settings:

Pitch Bend Limiter
The range from 0 to 127 can be exceeded inside T². Even with a small PB-range larger ranges can be described. Pitch bend values may go up to 600 or become negative. The problem starts when a tone generator outside of T² listens to this data (there is no pitch bend higher than 127 outside Tango), but also if you want to work with features like Auto Tune in the Player or "Audio Pitch Range %" in the Modifier. This can lead to sometimes dramatic-sounding problems in the audio output of T².

Normalize Volume Controller to...
This is about the relationship between the velocity, i.e., the "built-in" volume of a note and the level of controller values that describe this volume, while a note is sounding. Very few audio notes (voice, winds, strings, etc.) reach their highest volume immediately at their beginning. It usually takes a few milliseconds, often more than 100, until the volume of a note unfolds completely. Now the volume is sent with Midi notes at the outset as "velocity", at a time where it is not really known yet. The same also applies to the note pitch.

This is one of the reasons for the delay which is used by Tango's Audio to Midi:
T² waits for some time before a volume settled. Then, the highest volume value (up to this point) is used to determine the velocity of the Midi note.

Let's say that this highest value was 60 (the range for velocity is, as elsewhere in the Midi language 0 to 127). A synthesizer would therefore play the note with velocity 60, just under half of the maximum velocity of 127. Now, during the first 20 or so milliseconds

follows the volume description via controller 2 (called “breath control”), which also reaches a maximum of 60.
The controller cannot amplify a note. 127 just means “No attenuation”. So, if it had gone up to 127 instead, the semi-loud Midi note at its loudest spot would have been played with its full "half" volume. Perfect!
In this case, however, it is further reduced in volume to about half (60 of 60), so that less than a quarter of the maximum volume, that is 30 (of 127) remains.
So, since the volume description should not be switched off (Crescendo, volume vibrato, etc. cannot be represented otherwise), with this setting the value gets scaled. 96 has proven to be a good value in order to keep a little headroom after the attack of a note for any subsequent volume changes. At the loudest part of the note there would be ¾ of 60 left, so the volume/velocity would still be 45.

Initial Volume Controller is Zero
As stated before, a velocity is associated with each Midi note message. If the volume of Midi notes is to be controlled exclusively by the breath controller (2) the velocity sensitivity in the connected synthesizer / sampler must be set to zero. Then, each note is played as if it had the maximum volume. On the other hand, their volume can be attenuated by the describing controller. There are two possibilities for the beginning of notes:

- The preset volume at start of each note is zero, so that a moment before the beginning of the note a controller with the value "zero" is sent to the tone generator. Then, the synth sound’s attack phase, which is often carefully programmed, may remain inaudible. Side effect: The timing of the note tends to become late, depending on the material.
  The volume only goes up a fraction of a second later.
- The first describing (non-zero) volume-message that normally follows the actual note-on message is anticipated up until right before that message. Then the original attack phase of the note can be heard. This option makes the sound harder, often more vivid and improves the perceived timing of the notes.
  On the other hand, the sound result does not exactly reflect the conditions T² has found in the original audio.

Audio Gate
If a microphone is connected, T² records everything.
But maybe, not everything that is audible is intended as input for the program. For example, monitor signals in a concert with Tango spill into the microphone. These are misunderstood as notes to be analyzed and thus end up in the Listener – where the Player regards them as input it can use.
To prevent these “strange feedback loops”, this gate is useful to define a threshold. Below this level audio signals are ignored and not stored in memory.

Restore Default Settings restores the settings of the audio area (on the right), that seemed to work best during my testing.

If you have a different setup, you can save everything with
Save Settings. It is automatically loaded at startup, independent of the selected Room, and it is valid for all A2M modules and Listeners.

Metronome

As is conventional in music, a Metronome is responsible for the definitions of tempo and meter. All parts of the program which have to know exactly when the next “One” is going to come, how many quarter notes per minute are being played, or whether a measure has eight or only seven eighth notes (and also whether that is true about the following measures), get this information over the black connection from the Metronome. If you want to use the Modifier to quantize notes or to synchronize other parts of T², you need a Metronome.

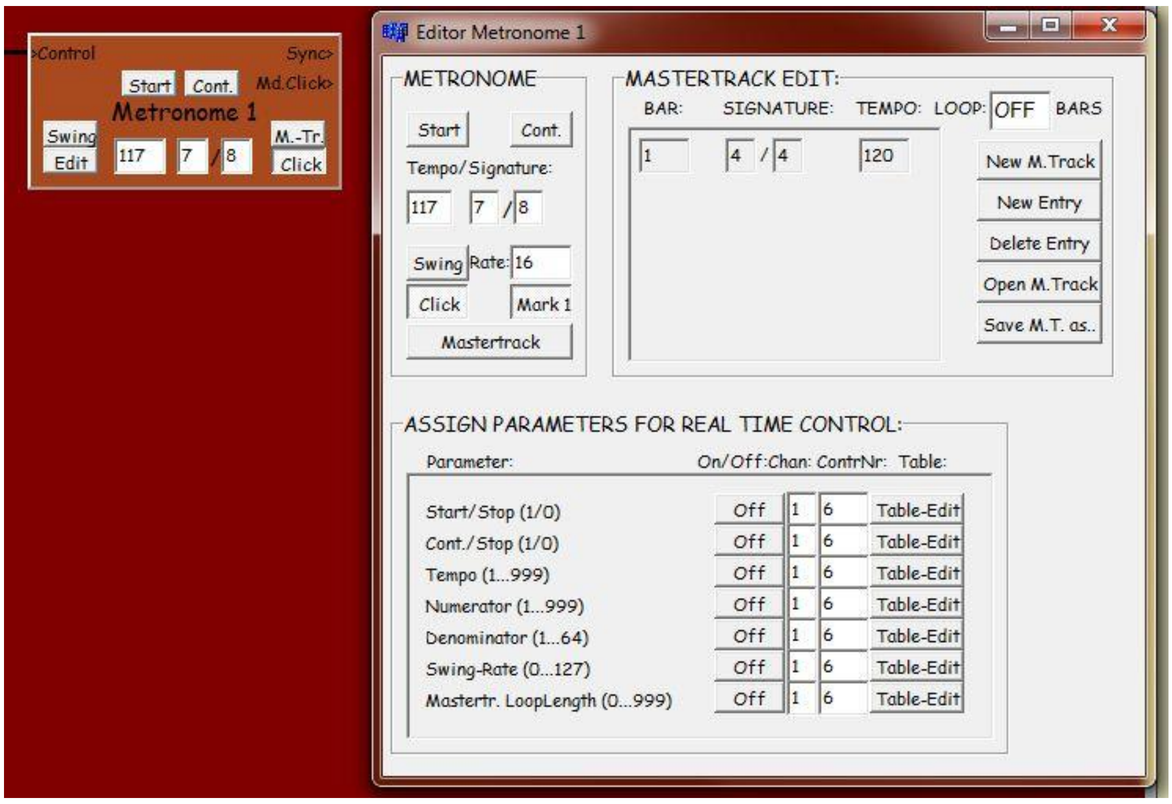
You can use several Metronomes so that T² plays in various tempos at once. When a Metronome is running, it knows the time of the next “One”, the time at which the next eighth note is due, or the time at which the next 32nd-note triplet will come (always in relation to the currently set tempo).

In the Room view it offers the control elements you would expect:

Start begins at bar 1, beat 1, while

Cont. (Continue) continues to count from whatever spot the Metronome was interrupted before.
Above, you can find out the current measure and quarter or eighth.

In the area below you can find tempo (bpm.) and meter as well as four buttons. Because all of them with the exception of “Edit” appear again in the Editor page, go directly there now by clicking on this button:



Metronome 1: 117 7 / 8

METRONOME: Tempo/Signature: 117 7 / 8; Rate: 16

MASTERTRACK EDIT: LOOP: OFF BARS

BAR:	SIGNATURE:	TEMPO:
1	4 / 4	120

ASSIGN PARAMETERS FOR REAL TIME CONTROL:

Parameter:	On/Off:	Chan:	ContrNr:	Table:
Start/Stop (1/0)	Off	1	6	Table-Edit
Cont./Stop (1/0)	Off	1	6	Table-Edit
Tempo (1...999)	Off	1	6	Table-Edit
Numerator (1...999)	Off	1	6	Table-Edit
Denominator (1...64)	Off	1	6	Table-Edit
Swing-Rate (0...127)	Off	1	6	Table-Edit
Mastertr. LoopLength (0...999)	Off	1	6	Table-Edit

The Metronome Editor

To the upper left are the actual Metronome functions:

Swing: As with normal sequencers, the moment of the offbeat (eighths between the quarters) can be delayed.

Rate: A value of 100 shifts it already very close to the next quarter note. “0” signifies “No Swing”

The **Click** can be turned on and off. Audio- as well as Midi- click are available. Midi-click is a Midi controller 1 with the value 127 on “One” of each bar and the value 64 for all other beats on Midi channel 1.

Mark 1: If turned off, “Ones” do not sound different than other clicks.

Mastertrack, as with other music programs, this turns a pre-programmed sequence of tempi and meter on or off. To the right of that you will find the

Editor of the Mastertrack, where you can edit tempos and/or meters for a particular position (measure number) of a Mastertrack. Before you can edit of course, you first have to generate a

New Entry. Mastertracks are separately saved in the Mastertracks folder, but when a Room is saved, the edited Mastertrack is always saved and brought up again with the Metronome. This causes the Metronome to always appear as it did at the moment when the Room was saved – just as with Tables.

Loop gives you the possibility of defining a loop in which a Mastertrack runs. If you need a pattern for example, that is made up of seven 4/4 measures and a following 3/4 measure, you would define a loop of 8 measures and generate an entry that sets measure 8 to 3/4. Or just open the Mastertrack “LoopExample.mtr” with

Open M. Track.

The Mastertrack only works when the “Mastertrack” button is pressed.

Below you can find almost all the parameters of the Metronome again, with the ability to be controlled remotely if you activate the Remote function here and join a green wire in the Room view with the control input. Be sure to pay attention to the Midi channel and control number.

Connections that control the Metronome by RTC are green. Synchronization connections originating from the Metronome or ending in the Metronome (synchronize it) are black.

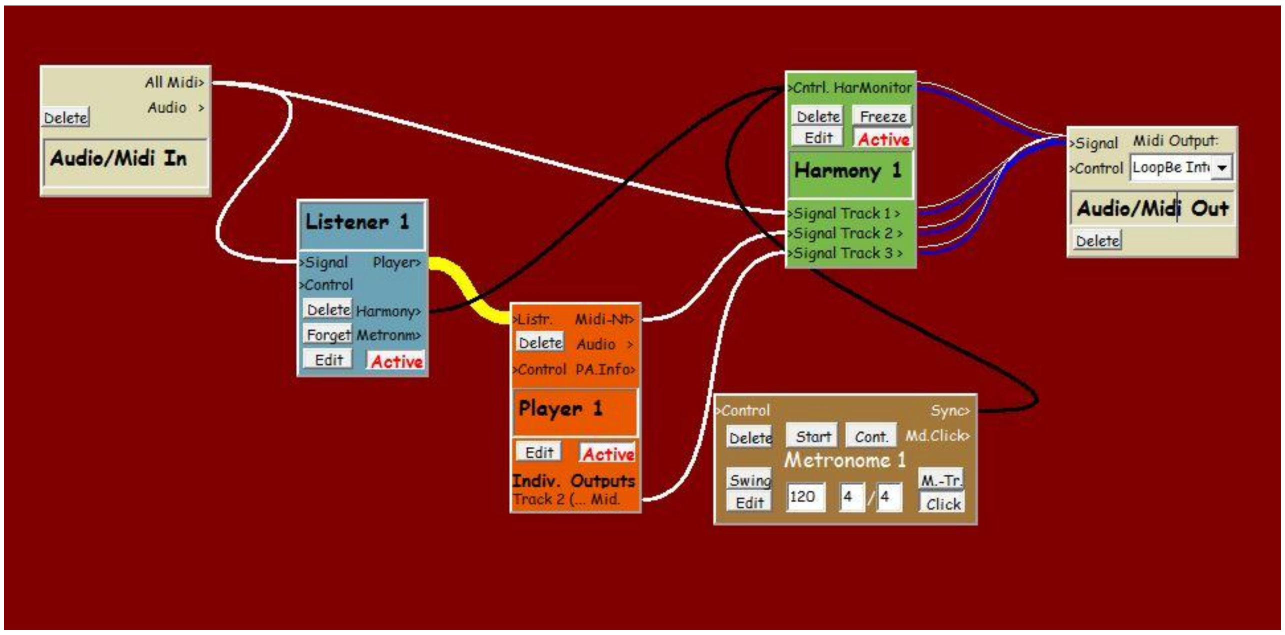
The Tables are modifiable, as always. For Start/Stop, the table displays “1” (for “On”) for the values over 63; otherwise it displays “0” (for “Off”).

If you play a tempo that a Listener hears and identifies, the Metronome can start and stop after it has taken over the tempo and/or the meter from you. It too needs a wire for this purpose (in this case, black) from the Listener’s Metronome output to the Control input of the Metronome.

The GUI of the RTC System in the Metronome is old and a bit outdated. However, it works like in the other modules. One of the next Versions of Tango will take care of this problem.

The Metronome can be synchronized from the outside world via Midi Clock. Just connect an Audio/Midi In module to the Metronomes control input.

Harmony



The Basic Function

The module’s purpose is to bend music, which is played by the Player without harmonic “objectives” so that the result will sound harmonically organized. In addition to the Player, Harmony’s input can also come in real time from Midi In. The signal reaches Harmony on the left at the input of a Harmony-track and leaves it corrected at its output on the right.

At any time there is a list of possible next pitches due to the current harmonic situation. From this list, a pitch for the next note on that track is selected according to certain criteria. Several tracks can be configured with different harmonizing rules.

In this process, the interval (third, fifth, etc.) and its direction (up or down) between the previously played and the current note can be preserved or, if necessary, altered as little as possible. Melodic material and chords can be processed differently. Tracks can also select from possible pitches in terms of their different "Harmonic Relevance". Here, it is possible to hear only chordal roots and fifths from a bass track, except if the notes are very quiet, very short or far away from the "one" of the bar.

There are two basic modes in which Harmony finds the "possible next pitches":

Floating Tonality

There is no "Current Chord" here, i.e. no currently valid chord name such as "G7" or "Cm". Instead, Harmony looks for the currently sounding pitches and creates a list of pitches that would be consonant with these. Unison, octaves, fifths (for diminished fifths please read the section on "Tritones in Float Tonality" below), fourths, major and minor thirds and sixths and their extensions into the next octave such as tenths are considered consonant. All the others are dissonant.

The name „Floating Tonality“ comes from the fact that in this mode the tonality of the music constantly changes (floats), due to the constant change of current sounding pitches. This effect is particularly evident when not all input notes begin and end at the same time (i.e. in polyphonic music).

Additionally, it is possible to use a musical voice (e.g. the live signal, as it comes from you via Midi In) as a **Cantus Firmus** which, although considered in the evaluation of currently sounding pitches, will not be corrected. Thus, tonality keeps floating, but only around the immovable fixed point of the notes played in real time by the user.

Defined Tonality

Here a "Current Chord" is defined in every moment using the corresponding scale to determine the possible next pitches.
There are three possible sources for the definition of a "Current Chord":

- **Leadsheet:** Chord symbols are assigned to the individual bars of a harmonic form. After pressing "Start", this chord sequence runs in a loop and sets a new "Current Chord" for each bar, which finally is used by the tracks to correct the input signal. In the input dialog "Select / Edit Chords" you can configure and save your own assignments of chord symbols, chords and scales. You can also work with several Leadsheets, e.g. if you like to vary the complexity of chords and scales used during an improvisation.

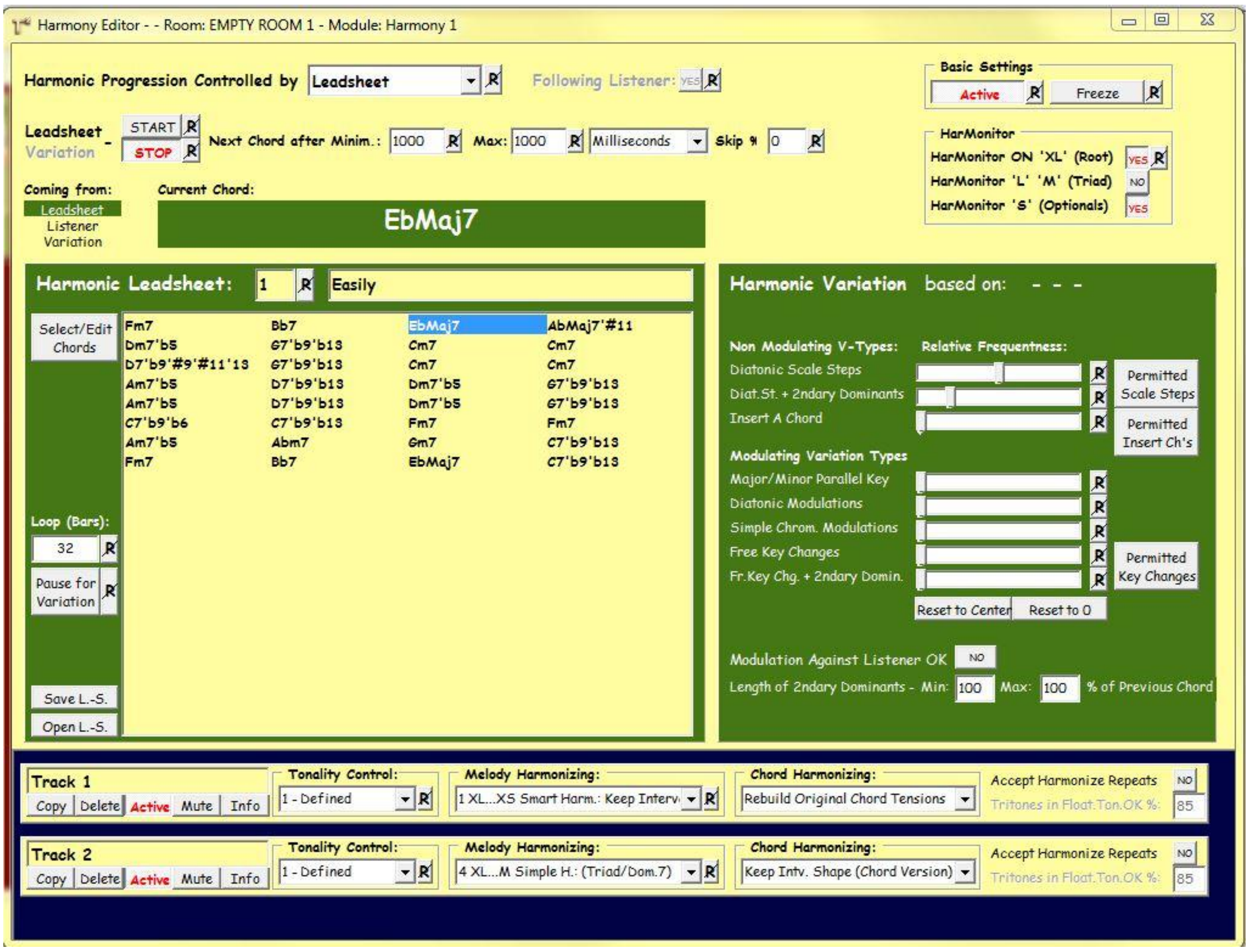
- The **Listener** constantly makes a harmonic analysis of your playing, both in terms of melody and chords. It can control Harmony with this information. Thus you can determine with your own harmony how T² harmonizes its answers to your playing.

- The **Variation-**function of Harmony can vary a "Current Chord" coming from the Leadsheet or Listener according to your wishes and thus constantly "invent" chords and chord progressions. With the control bars in the Variation area you determine the likelihood with which a specific kind of Variation is used at the next chord change.

  This **Variation** is also available as a ”chord type” for single Leadsheet- bars.
  The Leadsheet’s Variation is triggered by just entering "Variation" instead of a chord symbol for a particular bar. In the Leadsheet the preceding defined bar is varied, whilst in the case of Listener control, the chord the listener previously heard in your playing, will be varied. Variation differentiates between major, minor and dominant situations and behaves accordingly. The Variation is useful, in giving T² some harmonic freedom within your pre-determined limits in a given harmonic form.

HarMonitor plays the current "Current Chord" for your information. The function has not been fully implemented yet.

So much for the basic design of Harmony.


Harmony Editor - - Room: EMPTY ROOM 1 - Module: Harmony 1

Harmonic Progression Controlled by: Leadsheet

Following Listener: YES

Basic Settings: Active | Freeze

Leadsheet / Variation: START | STOP

Next Chord after Minim.: 1000 Max: 1000 Milliseconds Skip %: 0

HarMonitor
HarMonitor ON 'XL' (Root): YES
HarMonitor 'L' 'M' (Triad): NO
HarMonitor 'S' (Optionals): YES

Coming from: Leadsheet / Listener / Variation

Current Chord: EbMaj7

Harmonic Leadsheet: 1 Easily

Select/Edit Chords

Fm7	Bb7	EbMaj7	AbMaj7'#11
Dm7'b5	G7'b9'b13	Cm7	Cm7
D7'b9'#9'#11'13	G7'b9'b13	Cm7	Cm7
Am7'b5	D7'b9'b13	Dm7'b5	G7'b9'b13
Am7'b5	D7'b9'b13	Dm7'b5	G7'b9'b13
C7'b9'b6	C7'b9'b13	Fm7	Fm7
Am7'b5	Abm7	Gm7	C7'b9'b13
Fm7	Bb7	EbMaj7	C7'b9'b13

Loop (Bars): 32

Pause for Variation

Save L.-S. | Open L.-S.

Harmonic Variation based on: - - -

Non Modulating V-Types: Relative Frequentness:
Diatonic Scale Steps
Diat.St. + 2ndary Dominants
Insert A Chord

Permitted Scale Steps | Permitted Insert Ch's

Modulating Variation Types
Major/Minor Parallel Key
Diatonic Modulations
Simple Chrom. Modulations
Free Key Changes
Fr.Key Chg. + 2ndary Domin.

Permitted Key Changes

Reset to Center | Reset to 0

Modulation Against Listener OK: NO

Length of 2ndary Dominants - Min: 100 Max: 100 % of Previous Chord

Track 1 — Copy | Delete | Active | Mute | Info — Tonality Control: 1 - Defined — Melody Harmonizing: 1 XL...XS Smart Harm.: Keep Interv — Chord Harmonizing: Rebuild Original Chord Tensions — Accept Harmonize Repeats: NO — Tritones in Float.Ton.OK %: 85

Track 2 — Copy | Delete | Active | Mute | Info — Tonality Control: 1 - Defined — Melody Harmonizing: 4 XL...M Simple H.: (Triad/Dom.7) — Chord Harmonizing: Keep Intv. Shape (Chord Version) — Accept Harmonize Repeats: NO — Tritones in Float.Ton.OK %: 85

Harmony Edit

In the upper part the most important decisions can be found, as well as the basic transport functions of the harmonic progression and, if necessary, synchronization of Harmony.

Basic Settings are the buttons for Active (Harmony actually alters its input or not?) and Freeze (of the "Current Chord").

Harmonic Progression Controlled by Leadsheet or Listener with Variation (if required) answers the fundamental question of who has the say in harmony.

Following Listener: Even if in the previous parameter "Listener / Variation" was selected, it can be useful to have Harmony follow the Listener only temporarily or on special request. In the meantime, for example, the Variation can provide for a little variety in harmonic action. I am using an RTC switch to activate the Follow function and then leave it to the Listener to switch it off shortly after the end of my phrase, using the listener parameter "Current Rest". At this point the Variation begins to work until my next RTC "Follow" instruction. The result is less nervous, harmonically, than it would be with "Follow" permanently active.

If you want Harmony to be controlled by the Listener, you have to connect the Listener to Harmony. It will then be able to react to single notes heard by the Listener in order to avoid certain harmonic clashes between your playing and Variation-chords.

The Listener can hear harmony in your melodies as well as in chords you play.

Leadsheet (or) Variation Start / Stop: How often are chord changes supposed to happen?

Here, Harmony can be granted a certain freedom or a fixed value can be set. If you select “On Cue” here, this function will switch back to “Stop” after each new chord (and wait for your next cue).

As a unit for set values, both milliseconds and bars or different note values can be used. If you do not choose milliseconds as a unit, of course a connected metronome is required to synchronize Harmony. You can also specify

Skip % to omit a chord change now and then.

Most of these parameters can be controlled via RTC. For Start and Stop, there are two different buttons, so you can define different turn on and -off points for the harmonic "transport" in the RTC-Tables.

Start / Stop relates only to the Leadsheet and Variation and not to the control by the Listener, because only the timing of the first two functions are determined by T² whilst the Listener obviously hears a new chord when you play it.

HarMonitor provides a simple Midi version of the "Current Chord" on MIDI channel 1, wherein you are currently only able to determine whether only the root (actually only notes with the “Harmonic Relevance XL” – see below), only the triad (XL, L und M-notes) or a further optional chord note (“Harmonic Relevance S”) is played. HarMonitor is only a temporary, simple version of a future larger Harmony-chord player.

There is a special output for HarMonitor on the Room view of Harmony.

Current Chord represents the currently valid chord symbol, as Harmony's tracks use it.

To the left of the chord symbol you will find an indication of the source of the "Current Chord". It is often useful to know whether a particular chord has come from the Leadsheet, the Listener or from the Variation.

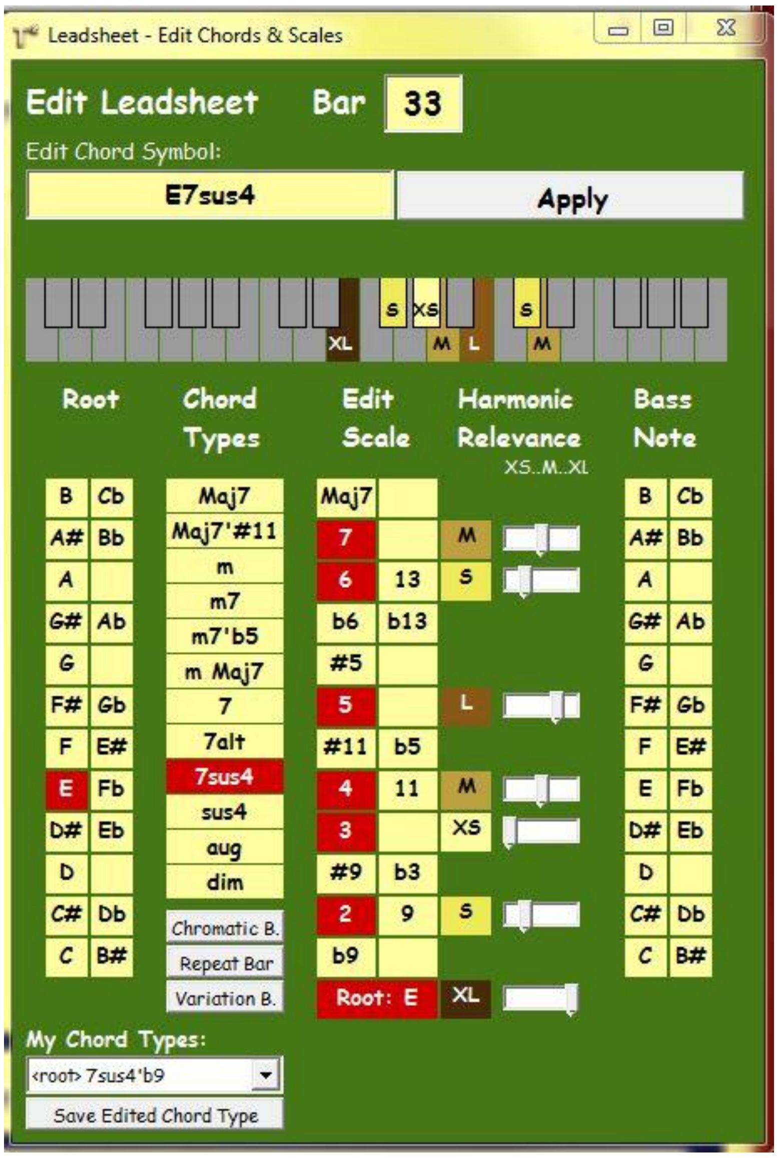
With the two areas below you can edit the Leadsheet and on the right the Variation parameters.

Leadsheet Editor

Here, it is possible to define a harmonic sequence which can be repeated as desired in a loop. There are 10 definable Leadsheets. It is possible to switch back and forth between them during operation, also via RTC.

The most important element is the editor which can be opened with the button “Select/Edit Chords”:

Edit Chords & Scales


Leadsheet - Edit Chords & Scales
Edit Leadsheet
Bar 33
Edit Chord Symbol:
E7sus4
Apply
S XS S
XL M L M
Root
Chord Types
Edit Scale
Harmonic Relevance
XS..M..XL
Bass Note
B Cb A# Bb A G# Ab G F# Gb F E# E Fb D# Eb D C# Db C B#
Maj7 Maj7'#11 m m7 m7'b5 m Maj7 7 7alt 7sus4 sus4 aug dim
Chromatic B. Repeat Bar Variation B.
Maj7 7 M 6 13 S b6 b13 #5 5 L #11 b5 4 11 M 3 XS #9 b3 2 9 S b9 Root: E XL
My Chord Types:
<root> 7sus4'b9
Save Edited Chord Type

Below the symbolic piano keyboard you can select chord symbols using root, chord type and, if required, a different bass note instead of the root of the chord.

With **Edit Scale,** you can change the scale associated with the chord types. You can take away or add notes, or alter intervals of the scale (i.e. make them a half step higher or lower). First, Harmony will assume that you wanted to replace one interval by another, but if you go on editing now, you can also add notes. Thus, scales with more or less than the normal 7 notes are possible. For each semitone above the root you will find one line, only because of the special harmonic implications of #5 (augmented fifth), I had to add an additional line there.

Harmony will use only the notes which are marked in red here.

The chord symbol appears in the box at the top („**Edit Chord Symbol**“) and is inserted at the current cursor position in the Leadsheet with the button

Apply.

Harmony attempts to arrive at a correct chord symbol based on your changes in the scale, but if you do not agree, you can edit the chord symbol.

On the piano keyboard, the notes of the scale are marked in color between dark and light brown. The colors and letters on the keys relate to the

Harmonic Relevance of the individual notes, which you can edit in the column below.

The tonal importance of a scale note is described in 5 classes (XS, S, M, L, XL, like T-shirts). If you only assign XL to a chord's root, it is (e.g.) possible to have a Harmony track, which is responsible for the bass, always look for the next root note of the current chord while it ignores all other scale notes. Therefore a setting called "XL   Simple Harmonize: (Root)" can be found in the Track parameters and in the HarMonitor, which also selects its notes with this system.

If you have, for example, marked both remaining triad notes (third and fifth) with an L, a track that only wants to play triads just has to look for the notes with L or with higher levels. In this case the root part of the triad was already marked XL, so triadic notes are marked as L and XL. With this system you can define simple scales as well as more unusual ones, relate them to chords and chord symbols and, finally, save and retrieve them quickly. The notes whose use you want to allow only as passing tones or suspensions, as they are less important or at times even disturbing for the tonality (such as the fourth above the tonic major key), are simply marked XS.

At the bottom of the Chords and Scales Editor, under

My Chord Types you will find the possibility to open and save chord types, -scales and -symbols that you have defined yourself.

All .chrd files in the folder ChordTypes are opened at startup. They are available in every Harmony module.

Below the chord types there are more controls for the Leadsheet:

Chromatic Bar adds a bar to the Leadsheet in which no chord is defined. The harmonic correction is canceled for this bar.

Repeat Bar repeats the previous bar of the Leadsheet.

Variation Bar provides no harmonic information, but activates the Variation function for this particular bar. How Variation works is decided with the probability regulators in the Variation area. Variation is based on the last defined chord. You can shape your desired Variation types in the Variation area.

You can find more Leadsheet functions which do not have to do with entering chords and scales in the Leadsheet area of the Harmony editor:

Loop (bars) determines whether and after how many bars the Leadsheet goes back to the first measure. So (e.g.), you can go only through the first 4 or 8 bars of a Leadsheet the first time through and next time use the whole harmonic form. Instead of the value "0" for the setting you find "Once Only". This means that the Leadsheet will only run once and after the last bar it will set the transport switch above to "Stop". When you add bars to the Leadsheet, “Loop” is automatically incremented.

With **Pause for Variation,** you can stop a running harmonic form. In the resulting "harmonic" break the Variation can take over the chord changes, until you remove the harmonic pause. You will probably trigger this function during a concert with T² via RTC. The effect is similar like surprisingly coming to a clearing in the woods and taking a little rest. With the Variation settings you can determine light and colors of the clearing.

Finally, you can save and open Leadsheets as .lds files using the **Open** and **Save** buttons.

The Leadsheet offers the usual

Copy / Paste / Undo functions using Ctrl-C, Ctrl-V, Ctrl-Z and delete. Redo is possible (Shift / Ctrl-Z). Move the mouse over the “Select/Edit Chords”-button to find a list with the available shortcuts.

Drag & Drop, i.e. selecting and dragging text is not possible.

If you wish to check the scale underlying a particular chord again, double-click to a chord in the Leadsheet. This allows you to edit scale, chord symbol and relevance classes. After those changes you must re-“Apply” the changed chord to the Leadsheet.

Now to the area of

Harmonic Variation

First you can see which chord the Variation is based on. That is the last defined chord (by the Leadsheet or by the Listener).

The faders below indicate the relative frequentness with which the respective harmonic Variation will be used. These are probabilities. The actual action to be taken is determined randomly.

If only one of the faders is greater than zero, the appropriate action will therefore take place every time. If no fader is greater than zero, no Variation can occur.

Non Modulating Variation Types:

Diatonic Scale Steps: On almost every step of a diatonic major or minor scale (e.g. the white piano keys), triads or bigger chords can be formed with three or more notes. These are harmonically quite compatible with each other and can be freely combined (within limits). Also, it is fairly easy to improvise over it without the need to know all the details, just using the respective scale. A typical sequence of Variations (based on CMaj) here could go like this: CMaj, Am, Em, Dm, G7 ...

Which steps you permit for major or minor situations is defined in the list you can open with the button

Permitted Scale Steps. T² then rolls the dice on the basis of the allowed options.

Diatonic Steps + 2ndary Dominants: Before each of these steps, a secondary dominant can be inserted. The result sounds more elegant and less medieval. The basic key will not be left. This could, for example, lead to the sequence CMaj, E7, Am ...

Insert A Chord: Here chords can be inserted, which are not diatonically related to the original key. Intervals for this purpose are defined on the right via

Permitted Insert Chords, separately for major, minor and dominant chords. For example, if enabled in the dominant line +1, Harmony could now and then insert a Db7 chord (+1 semitone higher) in a C major situation before it goes back to C major or something akin. Jazz musicians call this a tritone substitution of the dominant – the Db7 is a substitute for G7 chord. In FMaj the inserted chord could be only a Gb7 chord because of the interval-oriented definition of this setting.

If we play however, in the region of Ab minor and in the “major”-line of the list "+6" is activated, a D major chord could be inserted, (as Ab plus 6 semitones results in a D). The prerequisite is that the Fader of this kind of Variation is greater than zero - and that you like this harmonic progression.

Modulating Variation Types:

Here you have to be a bit more careful while playing with T^2, because here the program may leave the tonal reference point - so to speak, the greater harmonic region – in short: it may modulate.

Major/Minor Parallel Key: The most harmless variant. Here Harmony changes only to the parallel major or minor key (e.g. from CMaj to Am).

Diatonic modulation: As stated before, on almost every step of a diatonic major or minor scale (e.g. the white piano keys), triads or bigger chords can be formed with three or more notes. Most of these triads can then be treated as a new tonic, i.e. the new tonal center (greater harmonic region).

Simple Chromatic Modulation: It is really "simple", because chromatic modulation can normally do much more. Here, only the fact is exploited that every minor triad can be the (minor) subdominant (IV. step) not just of a minor, but also of a major chord. Every major triad reversely can be the dominant of a major, but also of a minor chord, as dominants are always in a major key even if they are aimed at minor tonics. An F minor may be followed by a C major as well as a C minor and G major can prepare C minor or C major. With this little trick one modulation step can eliminate or add up to four accidentals.

Free Key Changes: Here it is possible to allow changes to distant, unrelated keys, via

Key Changes Permitted on the right. Again they are defined by semitone distances from the home key. The starting mode (minor/major) and key, as well as the target mode/key are considered. Therefore in the default settings a key change from C major to D major and D minor (+ 2, + 2 in the top two rows) is allowed whilst Harmony may change to Eb major and Eb minor (+3 is activated in two lower rows) if it starts with a C minor here.

Free Key Changes + 2ndary Dominants: The free key changes appear softer if they are prepared by a secondary dominant.

Reset to Center: All faders which are not in the zero position (far left) will be moved to the center. All others remain set to zero.

Reset to 0: All faders are reset to zero.

Modulation Against Listener OK: When modulating, Variation is permitted and Harmony at the same time is set to follow the Listener. It is possible that these two features will conflict with one another. This may be suppressed by giving the Listener the upper hand and thus keeping Harmony from modulating against the wishes of the human partner. This parameter is only relevant if Harmony is connected to a Listener.

Length of 2ndary Dominants: Defines the length of these dominants in comparison to the tonics. Only relevant if Harmony "thinks" in milliseconds, i.e. not in sync with a Metronome.

Harmony Tracks

Music that comes from the player or from live Midi In is corrected by Harmony while it is passing through the tracks. Several tracks can be configured, if different musical voices (such as bass tracks) should be treated differently. The prerequisite is that the data (notes) reaches Harmony via separate channels/wires. So, in this case you should use single Player outputs for the different Player-tracks.

On the left you find

Copy, Delete, Active, Mute and Info to deal with basic track handling. All of these functions work likewise throughout the program. Above that you can assign a name.

Tonality Control regulates the fundamental decisions about the way this Harmony track works. Please refer to the sections above on **"Defined"** versus **"Floating Tonality".**

- **Defined** should be selected if you want to work with a "Current Chord" – whichever Harmony function produces it.
- **Floating** works without a "Current Chord" by simply looking at the currently sounding notes and looking for consonant notes with these.
- **Cantus Firmus** treats the incoming notes as not alterable and therefore passes them on unchanged. The other "floating" tracks however, have to acknowledge the notes on this track and must not play anything dissonant with it.
- **No Harmonize** does not correct anything, the effect is the same as if the button on the left would read "Thru" instead of "Active". An important option in case you want to control this parameter with RTC.

Tracks distinguish between melodic and chordal Player input.

Melodic Harmonizing specifies how to work with melodic material:

XL...XS Smart Harmonize: Keep Intervals measures the intervals in the incoming melodies, be it from the Player or via Midi In in real time from you. It then maintains the basic interval's class (e.g., third, fifth) and its direction. The only way to correct the melody

harmonically is to change a minor third to a major third etc. In this mode of operation, the original material gets changed as little as possible.
The disadvantage is that *all* the notes of the scale are used, also the lesser characteristic for a given tonality (e.g., the fourth above the tonic major key). If there are a lot of chromatics in the original, there may also be a greater deviation of the pitch in comparison to the original pitches. A chromatic scale will be converted into a diatonic scale. So many intervals become larger because chromatic minor seconds become whole tones in a diatonic scale. This may lead to a stretching of the register. After rests in the Input these deviations are canceled as inconspicuously as possible.

So, the difference between “Smart” and “Simple Harmonize” is the way the original material gets corrected: “Simple H.” simply looks for the closest available “legal” pitch and uses that. “Smart H.” checks the interval between the last and the current *original* note. Then it chooses a pitch that forms the same interval-class (e.g. a third) between the last and the current *corrected* note.

XL...XS Simple Harmonize: (All Scale Notes) no longer measures the input intervals, but uses the next pitch of the currently valid scale that is marked with XS or higher (S, M, L, XL) – in this case that is *all* the notes in the scale. Here there are no register deviations, but intervention on the intervals of the original sounds a bit harder. You may also receive occasional note repetitions due to the harmonization. On this please read below "Accept Harmonize Repeats".

XL...S Simple Harmonize: (No 'Avoid' Notes) uses only the notes that you have marked with S or higher (that is M, L, XL). This could be, for example, the notes of a pentatonic scale. Here, Harmony would simply exclude the fourth in major scale, because you have marked it as an XS note. Even more repetitions due to harmonization may occur. Please read below "Accept Harmonize repeats".

XL...M Simple Harmonize: (Triad/Dom.7)
XL/L Simple Harmonize: (Root & Fifth)
XL Simple Harmonize: (Root)

Here, the same rules apply, accordingly.

The deciding factor are the properties you have assigned to the individual scale notes in terms of Harmonic Relevance (page “Edit Chords & Scales”) .

You will find numbers in front of the melodic harmonizing methods so that this function can be remote controlled using RTC. Thus, it is possible to harmonize various incoming notes differently, depending on their length, volume and location in time. There is also a sample Room on this in the “Technical demos” folder.

In the far right of each track there is a parameter called

Accept Harmonize Repeats.

When a Harmony-track only searches for root notes ("XL") and the original melody currently uses smaller intervals (e.g. seconds or thirds), often the next possible note for two consecutive (and different) original notes is the same as for the note before. The result will be a repetition of the root by Harmony, which obviously did not occur in the original. This can sound boring and can therefore be prevented by this parameter using octave leaps instead of the repetitions.

Below that you will find,

Tritones in Floating Tonality OK (%).

The tritone (whether as a diminished fifth or augmented fourth) sometimes does not sound clearly dissonant, as it can also consist of two stacked minor thirds (which in turn are consonant). On the other hand Harmony, when it is working in Floating Mode, tends to be a bit short of legal (consonant) pitches.

You can set a percentage of how often a tritone is considered consonant or dissonant whilst Harmony works with Floating Tonality (see Tonality Control on this).

Chord Harmonizing

Chords coming from the Player or from “Midi Chords” are specially marked and can therefore be treated differently than melodies from Harmony.

There are three algorithms:

Keep Intervallic Shape (Chord version) first analyzes the melodic line between the upper note of the last and that of the current chord. This ”melodic line” is corrected as described in "XS - Keep + Bend Intervallic Shape" for melodic material (see above).

The chord, which is “hanging” beneath the freshly harmonized upper note, is analyzed according to its intervals and now subtle changes to these intervals are applied while preserving the interval-classes (e.g., thirds, sixths). Everything is XS, i.e. all scale notes are used equally. Thus, the melodic context between successive chords is not abandoned and chords retain their original internal interval structure.

Rebuild Original Chord Tensions analyzes all incoming chords regarding triad structures, dissonance, register and additional optional notes and re-builds it from scratch in the target tonality. Contradictions and inconsistencies in the requirements are evaluated by weighing up possible mistakes. Subsequently, the chord with the smallest mistake is selected, similarly to the way Player Chords works. Here the corrected chord is very similar to the original but the melodic context of the top notes is sometimes abandoned.

There is also the option of simply using the melodic method „XS - Keep + Bend Intervallic Shape" or to leave chords uncorrected.

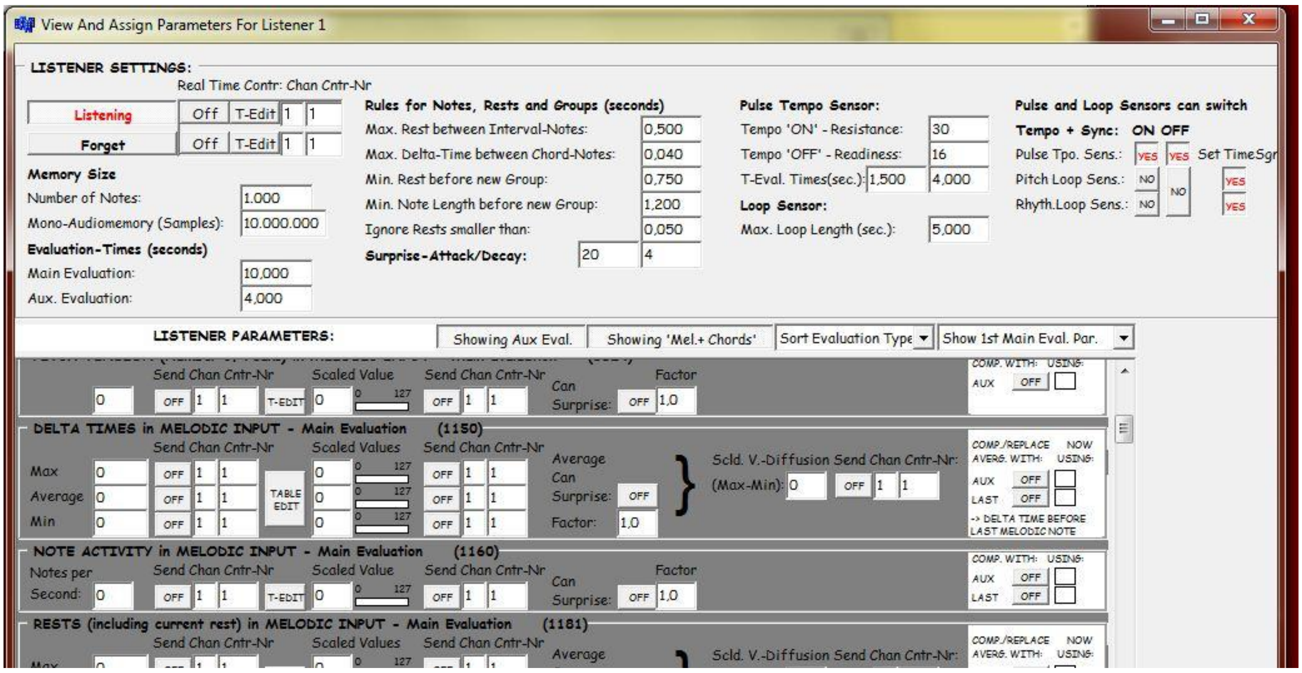
A last tip:

If you put Midi Chords between Midi In and Harmony, you can easily try out the chord functions in real time.

Listener

Basic Functionality

The Listener remembers the most recent music that reached it over its signal input – note for note. You yourself can determine the size of the memories for audio and Midi data. If the memory is full, it will begin to overwrite the data at the beginning, so that a trail of data for the Evaluation(s) is always available. If audio **and** Midi data arrive in the Listener, both are saved so that the relevant information regarding pitch and volume over the course of every individual note can be accessed at any time. Thus the audio memory, as opposed to other sampling memories, is exactly informed about the musical content of what is saved.



View And Assign Parameters For Listener 1

LISTENER SETTINGS:

Real Time Contr: Chan Cntr-Nr
Listening — Off — T-Edit — 1 — 1
Forget — Off — T-Edit — 1 — 1

Memory Size

Number of Notes:	1.000
Mono-Audiomemory (Samples):	10.000.000

Evaluation-Times (seconds)

Main Evaluation:	10,000
Aux. Evaluation:	4,000

Rules for Notes, Rests and Groups (seconds)

Max. Rest between Interval-Notes:	0,500
Max. Delta-Time between Chord-Notes:	0,040
Min. Rest before new Group:	0,750
Min. Note Length before new Group:	1,200
Ignore Rests smaller than:	0,050
Surprise-Attack/Decay:	20 — 4

Pulse Tempo Sensor:

Tempo 'ON' - Resistance:	30
Tempo 'OFF' - Readiness:	16
T-Eval. Times(sec.):	1,500 — 4,000

Loop Sensor:

Max. Loop Length (sec.):	5,000

Pulse and Loop Sensors can switch

Tempo + Sync: ON OFF
Pulse Tpo. Sens.: YES YES Set TimeSgr
Pitch Loop Sens.: NO — NO — YES
Rhyth.Loop Sens.: NO — YES

LISTENER PARAMETERS: Showing Aux Eval. | Showing 'Mel.+ Chords' | Sort Evaluation Type | Show 1st Main Eval. Par.

Send Chan Cntr-Nr | Scaled Value | Send Chan Cntr-Nr | Can Surprise | Factor
0 — OFF 1 1 — T-EDIT 0 (0–127) — OFF 1 1 — OFF — 1,0 — COMP. WITH: AUX OFF USING:

DELTA TIMES in MELODIC INPUT - Main Evaluation (1150)
Send Chan Cntr-Nr | Scaled Values | Send Chan Cntr-Nr
Max 0 — OFF 1 1 — 0 — OFF 1 1 — Average
Average 0 — OFF 1 1 — TABLE EDIT — 0 — OFF 1 1 — Can Surprise: OFF
Min 0 — OFF 1 1 — 0 — OFF 1 1 — Factor: 1,0
Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 0 — OFF 1 1
COMP./REPLACE AVERG. WITH: NOW USING: AUX OFF — LAST OFF — -> DELTA TIME BEFORE LAST MELODIC NOTE

NOTE ACTIVITY in MELODIC INPUT - Main Evaluation (1160)
Notes per Second: 0 — Send Chan Cntr-Nr OFF 1 1 — T-EDIT — Scaled Value 0 — Send Chan Cntr-Nr OFF 1 1 — Can Surprise: OFF — Factor 1,0
COMP. WITH: USING: AUX OFF — LAST OFF

RESTS (including current rest) in MELODIC INPUT - Main Evaluation (1181)
Send Chan Cntr-Nr | Scaled Values | Send Chan Cntr-Nr | Average | Scld. V.-Diffusion Send Chan Cntr-Nr: | COMP./REPLACE AVERG. WITH: NOW USING:

A glance at the editor of the Listener reveals two large areas:
Above, in the gray area, you can see in the Listener settings **whether** and **how** the Listener is listening. Here, rules are defined, memory space specified and the Evaluation (that is to say, the analysis of the notes saved in the memory) configured.

You can find the results of the Evaluation in the scrollable area below. There is a list here of about 200 Evaluation parameters, sortable and easy to find through shortcuts. These parameters describe what the Listener has heard.

*I will cover the control elements in the upper area, the **Listener settings**, when I talk about the Evaluation parameters which they configure. These sections are identifiable as an insertion in italics, as here.*

Listening and Not Listening *denotes whether the Listener is active or not. (The Listening button corresponds with the respective button in the Room view of the Listener like an "Active"- button.*
There, instead of "Not Listening" I use the shorter word "deaf". As long as the Listener is deaf, nothing is recorded or analyzed. Since there are no gradations between yes and no, the Table will say "no" (0) with input values of 0-63 and "yes" (1)

above 63. Activate RTC and connect a controller to the "Control" input of the Listener in the room view to switch the Listener on or off remotely.

This function was called "Freeze" in Tango I and in my concerts with the program it is always one of the really important real-time controls: When Tango plays something that I really like and I want the program to stay there for a while, I can freeze it. My playing has no effect on the program and thus T² steps into the background, becoming more like an accompaniment (instead of an unpredictable "counterpart").
The program keeps playing similarly to when I froze it, though without hiding completely behind repetitive loops. In the Player, there is a Freeze switch for each track with which it is possible to selectively use certain elements of Tango's music as background material and thus steer the attention of the audience towards me.

***Forget** makes the Listener erase its entire memory. The result is that the connected Player has nothing left to play and all Listener-timers (e.g. length of the current rest, etc.) and -parameters are at a standstill. Also this button you will find in the Room display again. If you have played something you would rather not have played (and to which T² is now responding), you can use this* to *get rid of it. After Forget the Listener will be waiting patiently for your next note.*
In the two music examples I used Forget, triggered by the *Double Click function to finish the two pieces without touching the mouse.*

***These two parameters, Listening and Forget, are the only Listener parameters which are controllable via RTC.** Otherwise the Listener in the RTC system is mainly used to **actively** control other modules or their parameters.*

Parameters like the first, "Pitch in Melodic Input – Main Evaluation (1110)," consider all notes in the memory that are not older than the number of seconds defined by the user in the Listener setting (Evaluation times, see below). Here **only the pitches** that are not parts of chords are considered. You can find the analysis of chord tones in another parameter, e.g. "Pitch in Chords – Main Evaluation (1210)."

*You can find the important **Listener settings** for the memory and the Evaluation(s) in the upper left of the Listener editor: the memory can be made larger or smaller for Midi with **Number of notes** and for the audio you can assign more memory space with **Mono-Audio Memory**.*

About sample space for audio:
With the default value of 10,000,000 samples, the Listener can save just about 4 minutes' worth of mono-sound. *You may think this seems like very little, but consider that rests are not included and the audio music saved here serves only as raw sound material for the Player module. Normally, 1000 notes and 10,000,000 samples are fully adequate.*

*In **Evaluation-Times** you specify how old notes may be before the Evaluation loses interest in them. Here, you define two memory caches, called "Main" and "Aux." Evaluation. Main Evaluation Time is preset to 10 seconds, Aux. to 4.*

Except for their Evaluation times, both Evaluations are identical. They use the Listener's memory, but not necessarily all of it.

Then why are there two Evaluations?

The reason lies in the fact that in musical contexts, 10 seconds are in some cases just right for the backwards analysis but in other cases too short or too long.

All these Evaluation parameters are not there, because I am interested in the statistics of my playing during the last couple of seconds. Instead, I want to use the changes in these values to influence the music T² plays, that is to control parameters, e.g. in the Player via RTC.

The length of the memory loop has an influence on the speed with which the average values change over time: The **longer** the loop, the **more** notes are part of an average and the **less** influence on this average has **one** single note, thus the **slower** the average changes. The average pitch value will change only rather slowly with a memory trail of 10 seconds, particularly when I have played maybe 30 or more notes within those 10 seconds.

If I need the average pitch value to change more quickly, I can use the second Evaluation, therefore called the “Aux.”-Evaluation.

You can find it further below in the light blue area, and the first parameter there is called “Pitch in Melodic Input – Aux.-Evaluation (3110).” Here the memory trail under consideration is preset to 4.0 seconds. The Aux.-Evaluation, in contrast to the Main-Evaluation, is marked in light blue (the dark gray-blue of the Main-Evaluation stands for a deeper look into the deeper, darker past).

> *As already stated, the memory length discussed here can be specified in the* ***Listener settings*** *with* ***Evaluation-Times****.*

If the short memory trail of the Aux.-Evaluation is still too cumbersome, you can also use an ultra-short memory. You can find the white area (white – no “depths of memory”) of the “Last Events” above the light blue parameters of the Aux.-Evaluation. There the 11th white parameter from above (“Last Melodic Pitch – Last Events 2110”) states the pitch of the **last**-played melody note. Of course no average can be built by one note; therefore these parameters hold only one line.

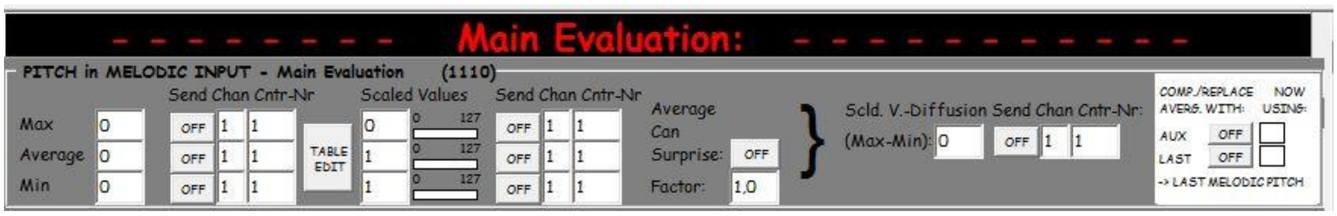
With this system you have the possibility to find the right mix of continuity (long memory, slow “musical breathing” and a stable performance by Tango) as well as quick reaction time (with a short memory) when it is needed for certain parameters.

In one of the next Tango versions there will be a “Control Manager” module for further processing of Listener messages, where many ways of cushioning, linkage and control of the controller streams will be encapsulated. Much on this topic can also be found in the folder "Useful Macros ". There I collected small groups of Modifiers to be able to perform certain more complex control functions with them. Explanations can be found again in the INFO of the Rooms.

The green parameter-area holds special parameters that do not fall into the three above-named Evaluations categories. More about this later.

The Control Elements of the Individual Evaluation Parameters

using the example of “**Pitch in Melodic Input – Main Evaluation (1110)**,” the first (top) parameter of the list:



A Midi-in connection from the keyboard should now exist. Call up the “Evaluation.room” to try out the following explanations and play a note on the keyboard. If nothing moves in the left column next to Max, Min and Average, something is not working with your Midi connection.

On the left you will find the words

Max, **Average** (as in, “average value”), **Min** and alongside those, three numerals. The parameter considers notes that are younger than (Main Evaluation) 10 seconds, analyses the highest and lowest note, makes these values known and ultimately builds an average value from all notes. These three “raw” values are presented in the column next to Max, Average and Min.

Of course it would be useless to know that the current average pitch is 62 (better known as middle D) if one cannot lead this information out of the Listener in order to control some function of the program with it. For that, the

Send-Buttons to the right side of the raw values are needed. If you click on the word OFF, this parameter’s value will be sent from the Listener via the Midi channel and controller on its right. Of course, this happens only if the value has changed at all and if the correct wiring in the Room display has been set up.

The function of the “Send” buttons – leading Evaluation parameters out of the Listener **in order to control other module with them:**
If you click on one of the buttons under the words “Send” in our example parameter, a few things happen in the user interface:

- On top of the parameter list (you must scroll up in order to see it) a yellow parameter is shown, which is an identical copy of this parameter. Yellow is the color of “active” parameters that actually are used to control other modules. For more clarity they are shown here in yellow, on top of the parameter list. You can make changes in this copy as well as in the original, as both versions will always appear the same, except for the color.

- In the Room display, where you can see the actual Listener module with its wiring, a new output which contains the parameters’ names hangs beneath the frame. If you move the mouse over this output, you can read its full name as well as information about the Midi channel and the control number of this Evaluation-Parameter in a tool tip.

If the value of a particular Evaluation parameter changes, this information will be sent by means of a Midi controller.

Do not forget to activate and adjust RTC for the desired parameter in target module (e.g. a Player).

We travel further from the left to the right in our example parameter:

All control elements right of the button **Table Edit** are sourced from **scaled values** instead of from the raw values of a parameter.

The logic of having Tables has been explained before. When that same logic is applied here, the area in which pitch normally ranges (about 30 to 90) is converted (scaled) so that the resulting value of the entire area ranges from 0 to 127. If you click on Table Edit, this is clearly illustrated by the graphic in the Table Editor on the lower right.

Going back again to the Listener page, you will see under the words "Scaled Values" the

scaled version of the raw values as well as their

graphic display, covering the range from 0 - 127.

You can find the above-mentioned "Send" options further to the right again, so that you can also send these values out of the Listener, if you want.

Average Can Surprise:
One Evaluation parameter (further below) has the name "Surprise." With this parameter, the program can be "startled" by sudden **changes** in the average parameters. You can activate as many parameters as you would like for the surprise function (OFF/ON) and specify the strength of the effect with the factors. Click on OFF and enter 2.0 as the factor.

Go to the right, to the box which reads "Show 1st Main Eval. Par." above the black bar ("- - - Main Evaluation: - - - -"). When you click on the box you will find "Show Surprise" in the pop-up menu that appears below. Go there and watch the graphic display of the surprise parameter whilst you play notes:

If you perpetually repeat the same middle C, the Listener will be surprised for a short time, though this surprise will quickly decrease. Now play other pitches and observe that the system answers the **changes in pitch-average** with an increasing amount of "Surprise."

Changes in the volume do not cause the same effect – as long as you don't activate Surprise for velocities as well.

Because the memory trail of the Main-Evaluation currently measures 10 seconds, at the latest, every "Surprise" disappears after this time span.

Because "Surprise" is a normal Evaluation parameter, you can also scale it with a Table and lead it out of the Listener with its own Send button, just as with every Listener parameter.

Those who are acquainted with Tango-1 will certainly be reminded of the TSI ("Time"-"Surprising Input") system and Quick Comment. All things said, the capabilities are now much more flexible and easily configured.

> *You can set the reaction time of "Surprise" in the* ***Listener settings*** *above with* ***Surprise-Attack/Decay****. "Attack" here means the rate of ascent and "Decay" the opposite.*

Now click on the popup menu, that reads "Show Surprise" and navigate via the upper entry of the pop-up ("Show 1st Main Eval. Par.") back to our pitch parameter.

This "Show..."-menu can help you searching for a particular Evaluation parameter from the existing 200 parameters.

To the right of the bracket, you will find

Scld. V.-Diffusion (Scaled Value Diffusion)
This parameter shows you the "amount of difference" between the scaled maximal and minimal values, which represents the statistical spread your play exhibits in regards to melodic pitch. If you repeat a note over a long period of time, the value here will be 0. In contrast, if you play from left to right over the entire width of the keyboard, this value will be nearer to 127. It is also strongly influenced by the length of the memory trail.

You will find the related "Send" button to its right.

The white area to the far right is only found in the Main-Evaluation:

Compare/Replace Average with...
Here, the results of the scaled values in the two other Evaluations (Aux.-Evaluation and Last Events) are compared with those of the main Evaluation, if you activate this comparison either for the Aux.-Evaluation or for the "Last Events".

Sounds complicated at first, I know. The idea behind it is as follows:

Because the Evaluation time (that is, the memory trail) of the Main-Evaluation is longer than that of the Aux.-Evaluation, processes such as the average value of the pitch move more sluggishly in the Main-Evaluation than in the Aux.-Evaluation.

For the Main Evaluation this makes sense, demanding a calmer reaction from T², avoiding hectic and sudden movements in reaction to my playing, and setting a certain stability to support my playing. This is exactly what I would demand from a fellow improviser. It better allows musical tension to develop over longer phrases without the music becoming too nervous.

On the other hand, a short Evaluation (Aux.-Evaluation or Last Events) by T² offers the possibility of a faster reaction from Tango to a sudden change of pace from me.

If you improvise with a human musical partner, you also have a certain "slower" memory, your "personal Main-Evaluation," that guides your playing and sets it in a context.

This can be overridden, however, by the actions of an improvising partner who suddenly interrupts the continuity, perhaps so much so that your whole focus is directed to this last unexpected phrase.

Exactly this sort of occurrence is meant to be simulated here: click next to the word “Last” on the button, so that you read “On.” Now play for about 10 seconds in the upper octave of your keyboard so that the average value is rather high. Then play **one** very low note. The comparison function of Last Events which you have just activated now copies the pitch value of this last note into the scaled average value of the Main Evaluation and all the older notes in the memory are ignored.

If the average value (which is much higher) is overridden by the last pitch, the field to the right of “On” is colored red. At the same time, you will see in the “scaled values” that the new minimum value (the pitch of the surprisingly low note) is copied into the average value. This happens however, only if the difference between the scaled average value and that of the new pitch is larger than 60 (in comparison with the Aux.-Evaluation 40); that is to say, if the last note makes a large enough jump beyond the bounds of coherence that had previously prevailed.

The “short circuit” is turned off as soon as the difference between the two Evaluations values is not larger than 5 points anymore. “Points” here always refer to the scaled values because only these are comparable.

Underneath in the white area you can find the name of the corresponding Last-Event-Parameter whose numbers start with a 2. The name of the commensurate Aux.-Evaluations-Parameter is identical to its respective main Evaluation parameter, except that the number begins with 3 (for example 3110) instead of with 1 (1110).

The Listener handles a total of about 200 parameters. In this jungle undergrowth of parameters, it can be useful to introduce some sort of order by …

Sorting the Evaluation Parameters

In the second pop-up menu from the right, over the individual Evaluation parameters, you can see the words…

Sort Evaluation Type.
If this sorting mode (Evaluation Type) is active and you scroll down through the long list of Evaluation parameters, you will find the following categories:

- Main Evaluation,
- Special Parameters,
- Last Events und
- Aux. Evaluation.

They are differentiated from one another through color (dark blue, green, white, and light blue) and category titles are marked in black bars.

Every one of these categories can be accessed quickly through the …

Show... menu to the far right:

Under “Show 1st Main Eval Par.,” which brings you to the first parameter of the Main Evaluation, you can find the entries that lead you to the first “Last Event Par.” and “Aux. Eval. Par.” Further below is the link to “1st Special Par.”

Now, in the “Sort” menu, choose …

Sort Event Type
Here the parameters will be sorted according to the musical context of the event. The new categories in the parameter list below are now:

- Melodic Input (monophonic single voices),
- Chords (notes that are played more or less at the same time and therefore considered part of a chord)
- Groups (parameters that describe Groups of notes and Group limitations) and
- All Input, where parameters are listed that don’t make a difference between melodic and chordal notes.

These categories can be found again in the “Show” menu as a link to its “1st Parameter.”

In the “Sort” menu when you choose

Sort Parameter Type, it will be sorted according to

- Special Parameters,
- Pitch,
- Intervals,
- Velocity,
- Length (note lengths are measured in seconds),
- Delta Times (the intervals of time between note beginnings, also measured in seconds),
- Note Activity (the amount of notes, mostly per second),
- Groups (Groups of notes and Group boundaries, as described above) and
- Rests (the rests between notes are measured partly in seconds, partly in their percentage of time allotment in the memory trail).

What belongs to each sorting category is coded in the *number* of each parameter: thousands stand for the Evaluation type, hundreds stand for the event type (melody, chord, etc.) and tens/ones stand for the parameter type (30 for example, has to do with volume). So when you look for the Aux.-Parameter which is identical with “Pitch in Melodic Input – Main Evaluation (1110),” you will want to look for Nr. 3110.

With the buttons

Showing Aux Eval and
Showing “Mel + Chords,” left of the “Sort” menu, you can hide all parameters of the Aux.-Evaluation, or all that describe both chords and melodic material. Because of its resulting smaller size, an overview of the Evaluation will then be easier to obtain.

For the next section, you should not hide the Aux.-Evaluation and the “Mel. And Chord-Parameters,” and you should select “Sort Evaluation Type” to the right of it.

The Parameters for the Main and Auxiliary Evaluations

All parameters which contain in their name
" ...in MELODIC INPUT," have to do with notes that are not a part of chords – with one exception: the highest note of every chord forms a melody with the highest note of the previous and following chords. This will be evaluated not only as a chord note but **also** as melody note.

> *As for the definition of "Chord": in the* ***Listener settings*** *you can specify how large the time span between chord notes may be (****Max. Delta-Time between Chord-Notes*** *– 40 milliseconds is the preset time span). All notes whose beginnings are nearer to each other belong to chords. The other notes are regarded as melody notes. You can find more on chords under "Chord Activity", preceding the section on Listener's "Special Parameters".*

Pitch: Midi considers pitches as numbers between 0 and 127, where 60 is defined as middle C. Many standard keyboards have only keys between 36 and 96, which is mostly adequate.

Velocity: means the volume of a note (actually, the speed with which the key is pressed down). A number between 1 and 127.

Length: The length specifications are adapted while long notes are still sounding. Play a few individual notes and then hold down a key for a longer time whilst watching the length parameter grow.

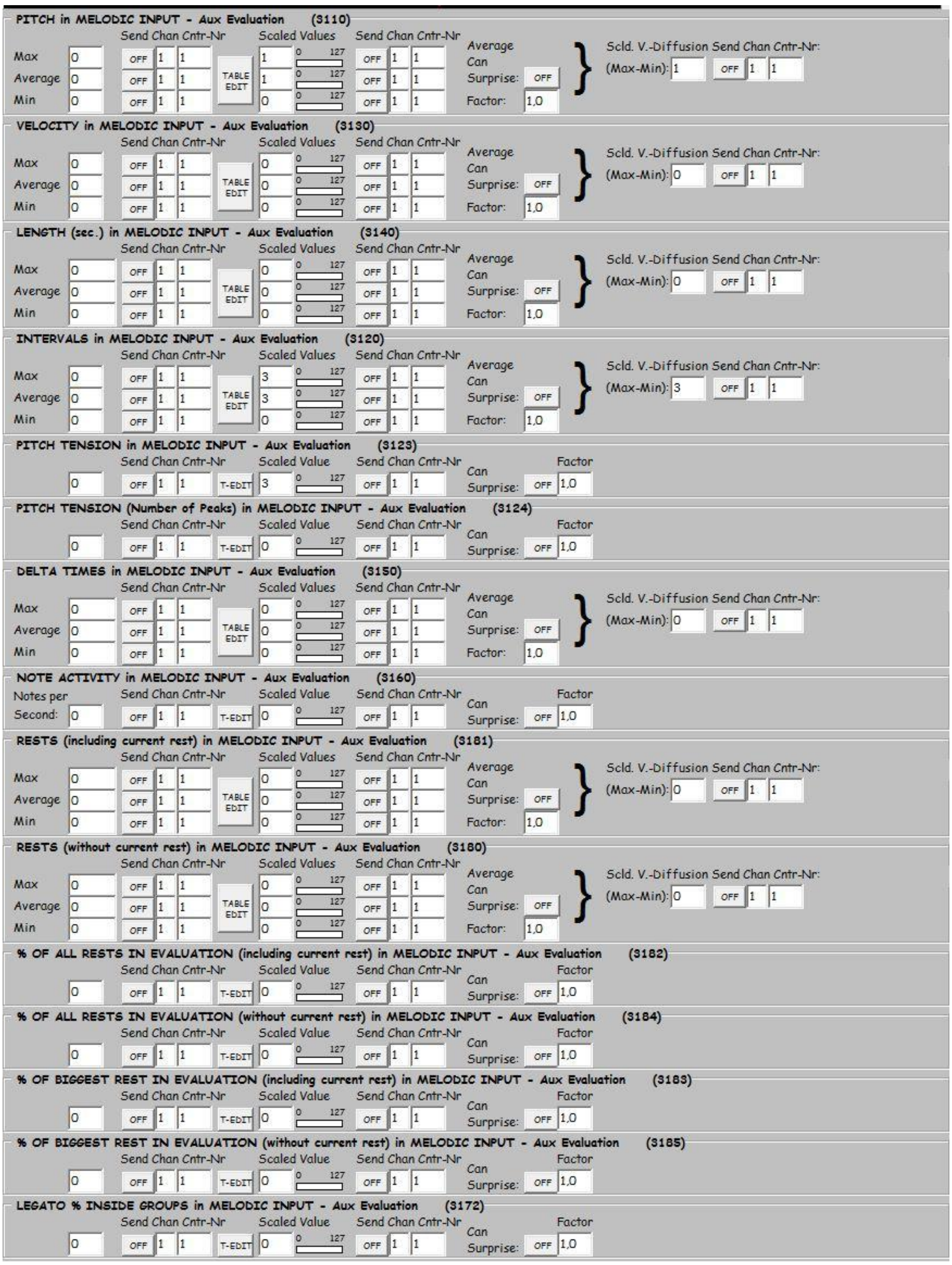
Interval: T^2 measures intervals in half steps, so a fifth is measured as a value of 7. In Main- and Aux.-Evaluation, this parameter does not differentiate between upwardly- and downwardly-constructed intervals, as no meaningful average can otherwise be formed.

Note: In contrast, Last Events distinguishes between "Last Melodic Interval (with direction)" and "Last Melodic Interval (size only)." The interval direction is accounted for in the first of the two parameters.

In chords, the notes are sorted initially by pitch, before the value for the individual intervals is determined.

If you look at the Table for intervals (button "Table Edit"), you will see how a value of 12 (an octave), which is rather large for a melodic interval, therefore will be scaled through the table up to a height of 96.

> *Notes separated by longer rests are heard by T^2 as belonging to different Groups and are thus not considered in the Evaluation as part of an interval. You can specify a maximum length for this rest between "interval notes" with* ***Max Rest between Interval-Notes*** *in the* ***Listener settings****.*



PITCH in MELODIC INPUT - Aux Evaluation (3110)

		Send	Chan	Cntr-Nr		Scaled Values		Send	Chan	Cntr-Nr
Max	0	OFF	1	1	TABLE EDIT	1	0 127	OFF	1	1
Average	0	OFF	1	1		1	0 127	OFF	1	1
Min	0	OFF	1	1		0	0 127	OFF	1	1

Average Can Surprise: OFF Factor: 1,0 } Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 1 OFF 1 1

VELOCITY in MELODIC INPUT - Aux Evaluation (3130)

		Send	Chan	Cntr-Nr		Scaled Values		Send	Chan	Cntr-Nr
Max	0	OFF	1	1	TABLE EDIT	0	0 127	OFF	1	1
Average	0	OFF	1	1		0	0 127	OFF	1	1
Min	0	OFF	1	1		0	0 127	OFF	1	1

Average Can Surprise: OFF Factor: 1,0 } Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 0 OFF 1 1

LENGTH (sec.) in MELODIC INPUT - Aux Evaluation (3140)

		Send	Chan	Cntr-Nr		Scaled Values		Send	Chan	Cntr-Nr
Max	0	OFF	1	1	TABLE EDIT	0	0 127	OFF	1	1
Average	0	OFF	1	1		0	0 127	OFF	1	1
Min	0	OFF	1	1		0	0 127	OFF	1	1

Average Can Surprise: OFF Factor: 1,0 } Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 0 OFF 1 1

INTERVALS in MELODIC INPUT - Aux Evaluation (3120)

		Send	Chan	Cntr-Nr		Scaled Values		Send	Chan	Cntr-Nr
Max	0	OFF	1	1	TABLE EDIT	3	0 127	OFF	1	1
Average	0	OFF	1	1		3	0 127	OFF	1	1
Min	0	OFF	1	1		0	0 127	OFF	1	1

Average Can Surprise: OFF Factor: 1,0 } Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 3 OFF 1 1

PITCH TENSION in MELODIC INPUT - Aux Evaluation (3123)

	Send	Chan	Cntr-Nr		Scaled Value		Send	Chan	Cntr-Nr	Can Surprise:	Factor
0	OFF	1	1	T-EDIT	3	0 127	OFF	1	1	OFF	1,0

PITCH TENSION (Number of Peaks) in MELODIC INPUT - Aux Evaluation (3124)

	Send	Chan	Cntr-Nr		Scaled Value		Send	Chan	Cntr-Nr	Can Surprise:	Factor
0	OFF	1	1	T-EDIT	0	0 127	OFF	1	1	OFF	1,0

DELTA TIMES in MELODIC INPUT - Aux Evaluation (3150)

		Send	Chan	Cntr-Nr		Scaled Values		Send	Chan	Cntr-Nr
Max	0	OFF	1	1	TABLE EDIT	0	0 127	OFF	1	1
Average	0	OFF	1	1		0	0 127	OFF	1	1
Min	0	OFF	1	1		0	0 127	OFF	1	1

Average Can Surprise: OFF Factor: 1,0 } Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 0 OFF 1 1

NOTE ACTIVITY in MELODIC INPUT - Aux Evaluation (3160)

		Send	Chan	Cntr-Nr		Scaled Value		Send	Chan	Cntr-Nr	Can Surprise:	Factor
Notes per Second:	0	OFF	1	1	T-EDIT	0	0 127	OFF	1	1	OFF	1,0

RESTS (including current rest) in MELODIC INPUT - Aux Evaluation (3181)

		Send	Chan	Cntr-Nr		Scaled Values		Send	Chan	Cntr-Nr
Max	0	OFF	1	1	TABLE EDIT	0	0 127	OFF	1	1
Average	0	OFF	1	1		0	0 127	OFF	1	1
Min	0	OFF	1	1		0	0 127	OFF	1	1

Average Can Surprise: OFF Factor: 1,0 } Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 0 OFF 1 1

RESTS (without current rest) in MELODIC INPUT - Aux Evaluation (3180)

		Send	Chan	Cntr-Nr		Scaled Values		Send	Chan	Cntr-Nr
Max	0	OFF	1	1	TABLE EDIT	0	0 127	OFF	1	1
Average	0	OFF	1	1		0	0 127	OFF	1	1
Min	0	OFF	1	1		0	0 127	OFF	1	1

Average Can Surprise: OFF Factor: 1,0 } Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 0 OFF 1 1

% OF ALL RESTS IN EVALUATION (including current rest) in MELODIC INPUT - Aux Evaluation (3182)

	Send	Chan	Cntr-Nr		Scaled Value		Send	Chan	Cntr-Nr	Can Surprise:	Factor
0	OFF	1	1	T-EDIT	0	0 127	OFF	1	1	OFF	1,0

% OF ALL RESTS IN EVALUATION (without current rest) in MELODIC INPUT - Aux Evaluation (3184)

	Send	Chan	Cntr-Nr		Scaled Value		Send	Chan	Cntr-Nr	Can Surprise:	Factor
0	OFF	1	1	T-EDIT	0	0 127	OFF	1	1	OFF	1,0

% OF BIGGEST REST IN EVALUATION (including current rest) in MELODIC INPUT - Aux Evaluation (3183)

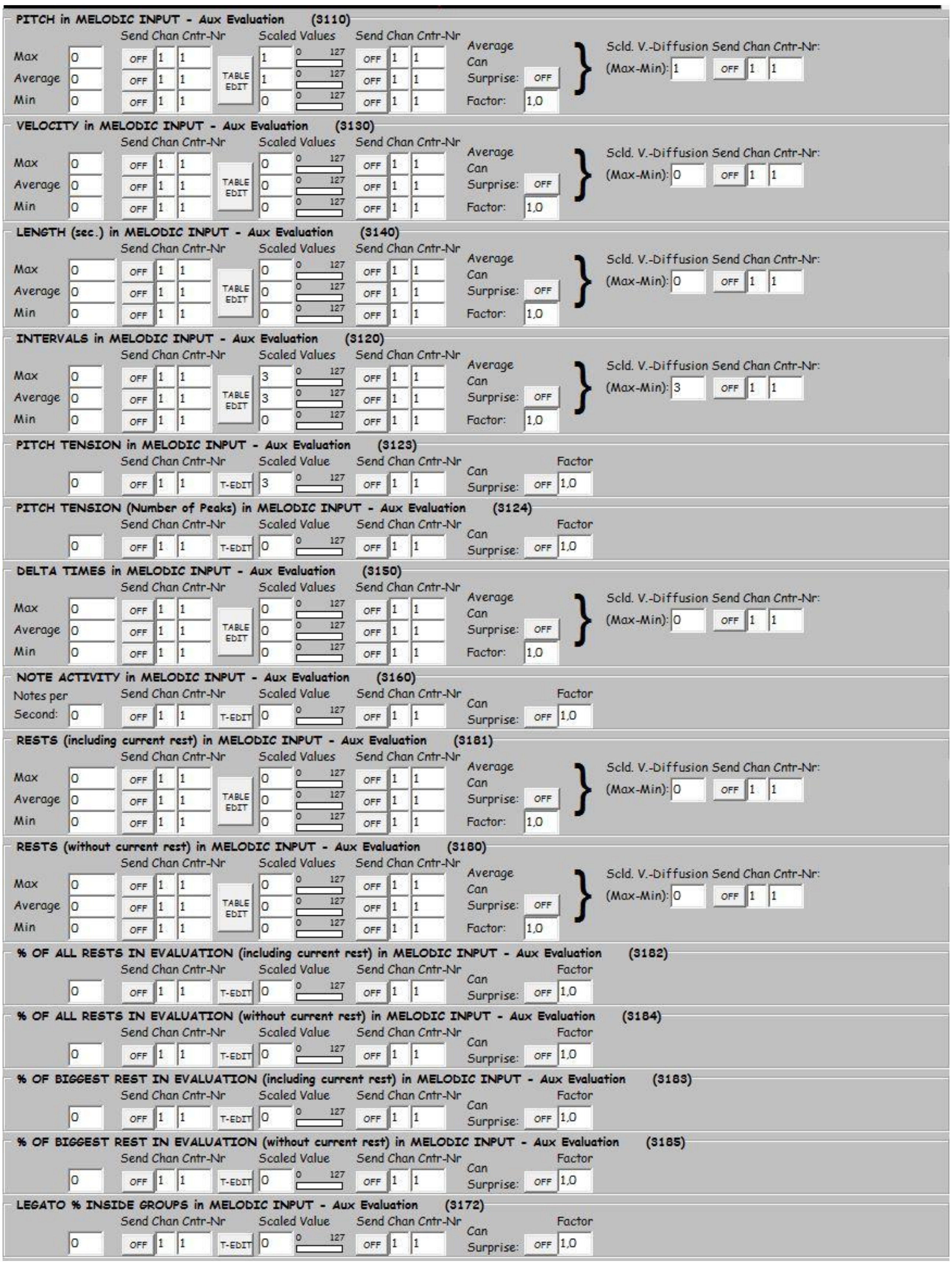
	Send	Chan	Cntr-Nr		Scaled Value		Send	Chan	Cntr-Nr	Can Surprise:	Factor
0	OFF	1	1	T-EDIT	0	0 127	OFF	1	1	OFF	1,0

% OF BIGGEST REST IN EVALUATION (without current rest) in MELODIC INPUT - Aux Evaluation (3185)

	Send	Chan	Cntr-Nr		Scaled Value		Send	Chan	Cntr-Nr	Can Surprise:	Factor
0	OFF	1	1	T-EDIT	0	0 127	OFF	1	1	OFF	1,0

LEGATO % INSIDE GROUPS in MELODIC INPUT - Aux Evaluation (3172)

	Send	Chan	Cntr-Nr		Scaled Value		Send	Chan	Cntr-Nr	Can Surprise:	Factor
0	OFF	1	1	T-EDIT	0	0 127	OFF	1	1	OFF	1,0



PITCH in MELODIC INPUT - Aux Evaluation (3110)

		Send	Chan	Cntr-Nr		Scaled Values		Send	Chan	Cntr-Nr
Max	0	OFF	1	1	TABLE EDIT	1	0 127	OFF	1	1
Average	0	OFF	1	1		1	0 127	OFF	1	1
Min	0	OFF	1	1		0	0 127	OFF	1	1

Average Can Surprise: OFF Factor: 1,0

Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 1 OFF 1 1

VELOCITY in MELODIC INPUT - Aux Evaluation (3130)

		Send	Chan	Cntr-Nr		Scaled Values		Send	Chan	Cntr-Nr
Max	0	OFF	1	1	TABLE EDIT	0	0 127	OFF	1	1
Average	0	OFF	1	1		0	0 127	OFF	1	1
Min	0	OFF	1	1		0	0 127	OFF	1	1

Average Can Surprise: OFF Factor: 1,0

Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 0 OFF 1 1

LENGTH (sec.) in MELODIC INPUT - Aux Evaluation (3140)

		Send	Chan	Cntr-Nr		Scaled Values		Send	Chan	Cntr-Nr
Max	0	OFF	1	1	TABLE EDIT	0	0 127	OFF	1	1
Average	0	OFF	1	1		0	0 127	OFF	1	1
Min	0	OFF	1	1		0	0 127	OFF	1	1

Average Can Surprise: OFF Factor: 1,0

Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 0 OFF 1 1

INTERVALS in MELODIC INPUT - Aux Evaluation (3120)

		Send	Chan	Cntr-Nr		Scaled Values		Send	Chan	Cntr-Nr
Max	0	OFF	1	1	TABLE EDIT	3	0 127	OFF	1	1
Average	0	OFF	1	1		3	0 127	OFF	1	1
Min	0	OFF	1	1		0	0 127	OFF	1	1

Average Can Surprise: OFF Factor: 1,0

Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 3 OFF 1 1

PITCH TENSION in MELODIC INPUT - Aux Evaluation (3123)

	Send	Chan	Cntr-Nr		Scaled Value		Send	Chan	Cntr-Nr	Can Surprise:	Factor
0	OFF	1	1	T-EDIT	3	0 127	OFF	1	1	OFF	1,0

PITCH TENSION (Number of Peaks) in MELODIC INPUT - Aux Evaluation (3124)

	Send	Chan	Cntr-Nr		Scaled Value		Send	Chan	Cntr-Nr	Can Surprise:	Factor
0	OFF	1	1	T-EDIT	0	0 127	OFF	1	1	OFF	1,0

DELTA TIMES in MELODIC INPUT - Aux Evaluation (3150)

		Send	Chan	Cntr-Nr		Scaled Values		Send	Chan	Cntr-Nr
Max	0	OFF	1	1	TABLE EDIT	0	0 127	OFF	1	1
Average	0	OFF	1	1		0	0 127	OFF	1	1
Min	0	OFF	1	1		0	0 127	OFF	1	1

Average Can Surprise: OFF Factor: 1,0

Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 0 OFF 1 1

NOTE ACTIVITY in MELODIC INPUT - Aux Evaluation (3160)

		Send	Chan	Cntr-Nr		Scaled Value		Send	Chan	Cntr-Nr	Can Surprise:	Factor
Notes per Second:	0	OFF	1	1	T-EDIT	0	0 127	OFF	1	1	OFF	1,0

RESTS (including current rest) in MELODIC INPUT - Aux Evaluation (3181)

		Send	Chan	Cntr-Nr		Scaled Values		Send	Chan	Cntr-Nr
Max	0	OFF	1	1	TABLE EDIT	0	0 127	OFF	1	1
Average	0	OFF	1	1		0	0 127	OFF	1	1
Min	0	OFF	1	1		0	0 127	OFF	1	1

Average Can Surprise: OFF Factor: 1,0

Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 0 OFF 1 1

RESTS (without current rest) in MELODIC INPUT - Aux Evaluation (3180)

		Send	Chan	Cntr-Nr		Scaled Values		Send	Chan	Cntr-Nr
Max	0	OFF	1	1	TABLE EDIT	0	0 127	OFF	1	1
Average	0	OFF	1	1		0	0 127	OFF	1	1
Min	0	OFF	1	1		0	0 127	OFF	1	1

Average Can Surprise: OFF Factor: 1,0

Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 0 OFF 1 1

% OF ALL RESTS IN EVALUATION (including current rest) in MELODIC INPUT - Aux Evaluation (3182)

	Send	Chan	Cntr-Nr		Scaled Value		Send	Chan	Cntr-Nr	Can Surprise:	Factor
0	OFF	1	1	T-EDIT	0	0 127	OFF	1	1	OFF	1,0

% OF ALL RESTS IN EVALUATION (without current rest) in MELODIC INPUT - Aux Evaluation (3184)

	Send	Chan	Cntr-Nr		Scaled Value		Send	Chan	Cntr-Nr	Can Surprise:	Factor
0	OFF	1	1	T-EDIT	0	0 127	OFF	1	1	OFF	1,0

% OF BIGGEST REST IN EVALUATION (including current rest) in MELODIC INPUT - Aux Evaluation (3183)

	Send	Chan	Cntr-Nr		Scaled Value		Send	Chan	Cntr-Nr	Can Surprise:	Factor
0	OFF	1	1	T-EDIT	0	0 127	OFF	1	1	OFF	1,0

% OF BIGGEST REST IN EVALUATION (without current rest) in MELODIC INPUT - Aux Evaluation (3185)

	Send	Chan	Cntr-Nr		Scaled Value		Send	Chan	Cntr-Nr	Can Surprise:	Factor
0	OFF	1	1	T-EDIT	0	0 127	OFF	1	1	OFF	1,0

LEGATO % INSIDE GROUPS in MELODIC INPUT - Aux Evaluation (3172)

	Send	Chan	Cntr-Nr		Scaled Value		Send	Chan	Cntr-Nr	Can Surprise:	Factor
0	OFF	1	1	T-EDIT	0	0 127	OFF	1	1	OFF	1,0

Pitch Tension: A common method in using pitch to create musical tension is the utilization of "peak" pitches (pitches surrounded by other lower-pitched notes). These prominent notes are heard as especially important and can structure a musical flow.

An especially large amount of energy and tension is formed when these higher pitches **form an ascending melody amongst themselves** (that is without the lower, surrounding ones).

The first of the two parameters describes how strongly of an ascending line this is while the second simply counts the number of peak tones in the memory trail. That is why this value in the Main-Evaluation is mostly larger than in the Aux.-Evaluation (more peaks).

Delta-Times designate the intervals of time (measured in seconds) between **the beginnings** of notes. Whether they are played staccato or legato is of no concern here.

Rests: This designates all parameters that evaluate rests between notes or after the last-played note. If some time has passed since the last-played note, at present there is logically a rest. Prior to this last (current) rest there were possibly others. At times the parameter "rest" is needed as a stable value providing information about the past relationship between notes and rests. In these cases the "Current Rest" will not be taken into account as its value would obviously be growing over time if you don't play. Again, sometimes it **is** important and must be looked at.

The current rest is included in parameters 1181, 1182 and 1183 (the maximum and average values are therefore always getting bigger here when you don't play) but not included in parameters 1180, 1184, and 1185.

The "Rest" parameters with percentage specifications refer to the entire memory trail (100%) as you defined them for the two Evaluations in Listener Settings/Evaluation-Times. You have the choice of evaluating the sum of all rests or only the largest rest.

> ***Smallest Rests***:
> *Often legato playing contains very small, unintended rests between the notes which, although practically inaudible, appear in the Rest parameters as part of the average or as "Minimum-Rest", therefore distorting the results. In the **Listener settings** you can prevent that with **Ignore Rests smaller than ….***

Note: The Tables with which time-related parameters such as Length, Rests or Delta Times are scaled are logarithmic so that the scaled numeral values will correspond to a human perception of time difference: in hearing, we take the difference between 500 and 1000 milliseconds more seriously than the difference between 5000 and 6000 milliseconds, although the latter is twice as large. Scaled with this logarithmic Table, the difference between 500 and 1000 milliseconds equals a value of "26," whereas the difference between 5000 and 6000 milliseconds amounts only to a "4."

Legato: Note lengths and rests between notes are compared here, but only inside a Group of notes that belong together, in other words, a phrase. Rests between various phrases in contrast, are longer and therefore not taken into account. Not only the rests are measured, but also the relationship between note lengths and following rests is considered. 50% means that notes and rests are equally long.

Group Parameters (1375, 1376 and 1363) specify the total number of Groups (or phrases) in the memory trail as well as the number of notes within them and the Groups' lengths in seconds. You can find out here if the input consists more of a long melody or of short, choppy statements.

> *You can define what the Listener understands by a Group in the* ***Listener settings*** *by* ***Min. Rest before a new Group****. Most of the time, a Group follows a rest and this rests' minimum length is set here. If you would also like a long note (as the closing note of a phrase) to be able to conclude a Group, then you can define its minimum note length in the parameter below –* ***Min. Note Length before new Group****.*

Note**: I use the terms Group and phrase mostly interchangeably (except when I talk about groups of modules). "Group" (capital G) is the technical term for the Listener parameter, "phrase" stands for the use the Player makes of these Groups.**

The Chord Parameters, in many cases, function like their respective melody counterparts. Here are just the exceptions:

Intervals are, as previously stated, ordered from top to bottom in chords.

Number of Notes specifies the size of each individual chord and

Width is the interval between the highest and lowest notes of a chord. The

Dissonance is not dependant on the amount of notes but instead on the interval structure within a chord. Sevenths and seconds are more dissonant than octaves, fifths or thirds.

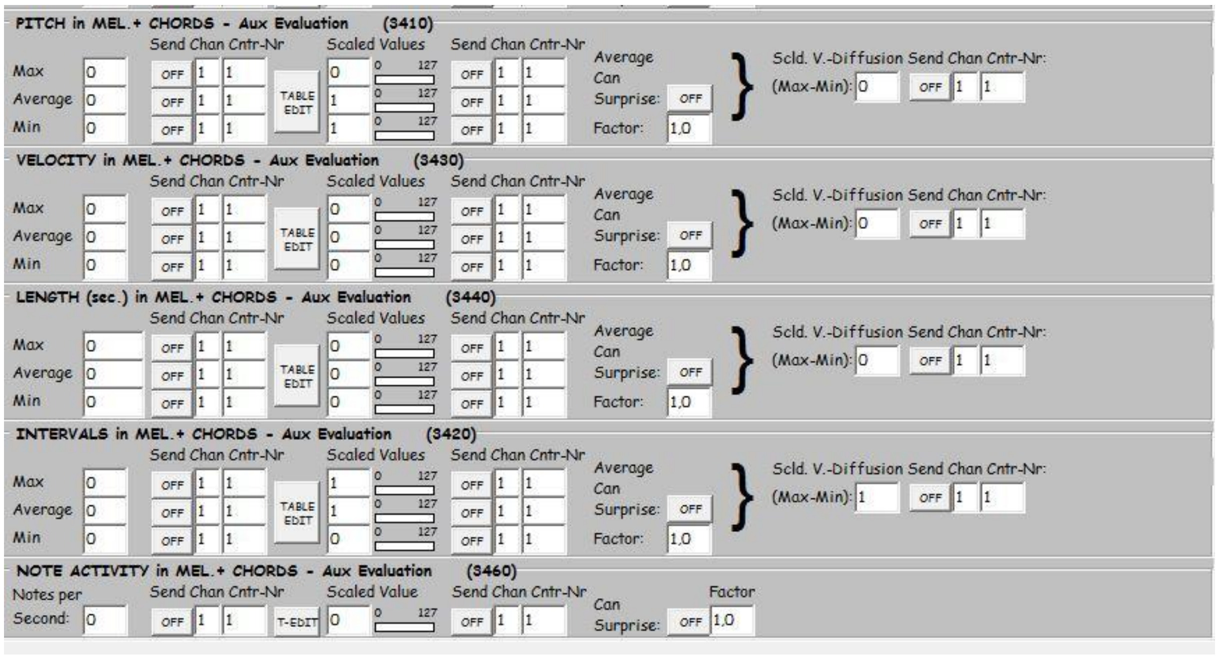
Pitch of Chord Top Notes evaluates the top chord notes, that is, the **most obvious** melodic part of chord sequences,

Delta-Times are the intervals of time between chords and

Chord Activity counts the chords in the memory trail.

> ***About the definition of "chord":***
> *Chords normally consist of several notes played at (about) the same time. What "about the same time" means, you can specify in the* ***Listener settings:***
> *How large may the largest interval of time between chord notes be (****Max. Delta-Time between Chord-Notes*** *– 40 milliseconds is the default value). All notes that lie further apart (time wise) do not belong to chords.*

The Parameters with “Mel + Chords” in their name evaluate all notes equally; that is to say, they measure both chord- and melody-notes.



PITCH in MEL.+ CHORDS - Aux Evaluation (S410)

		Send	Chan	Cntr-Nr		Scaled Values		Send	Chan	Cntr-Nr
Max	0	OFF	1	1	TABLE EDIT	0	0 127	OFF	1	1
Average	0	OFF	1	1		1	0 127	OFF	1	1
Min	0	OFF	1	1		1	0 127	OFF	1	1

Average Can Surprise: OFF
Factor: 1,0
Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 0 OFF 1 1

VELOCITY in MEL.+ CHORDS - Aux Evaluation (S430)

		Send	Chan	Cntr-Nr		Scaled Values		Send	Chan	Cntr-Nr
Max	0	OFF	1	1	TABLE EDIT	0	0 127	OFF	1	1
Average	0	OFF	1	1		0	0 127	OFF	1	1
Min	0	OFF	1	1		0	0 127	OFF	1	1

Average Can Surprise: OFF
Factor: 1,0
Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 0 OFF 1 1

LENGTH (sec.) in MEL.+ CHORDS - Aux Evaluation (S440)

		Send	Chan	Cntr-Nr		Scaled Values		Send	Chan	Cntr-Nr
Max	0	OFF	1	1	TABLE EDIT	0	0 127	OFF	1	1
Average	0	OFF	1	1		0	0 127	OFF	1	1
Min	0	OFF	1	1		0	0 127	OFF	1	1

Average Can Surprise: OFF
Factor: 1,0
Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 0 OFF 1 1

INTERVALS in MEL.+ CHORDS - Aux Evaluation (S420)

		Send	Chan	Cntr-Nr		Scaled Values		Send	Chan	Cntr-Nr
Max	0	OFF	1	1	TABLE EDIT	1	0 127	OFF	1	1
Average	0	OFF	1	1		1	0 127	OFF	1	1
Min	0	OFF	1	1		0	0 127	OFF	1	1

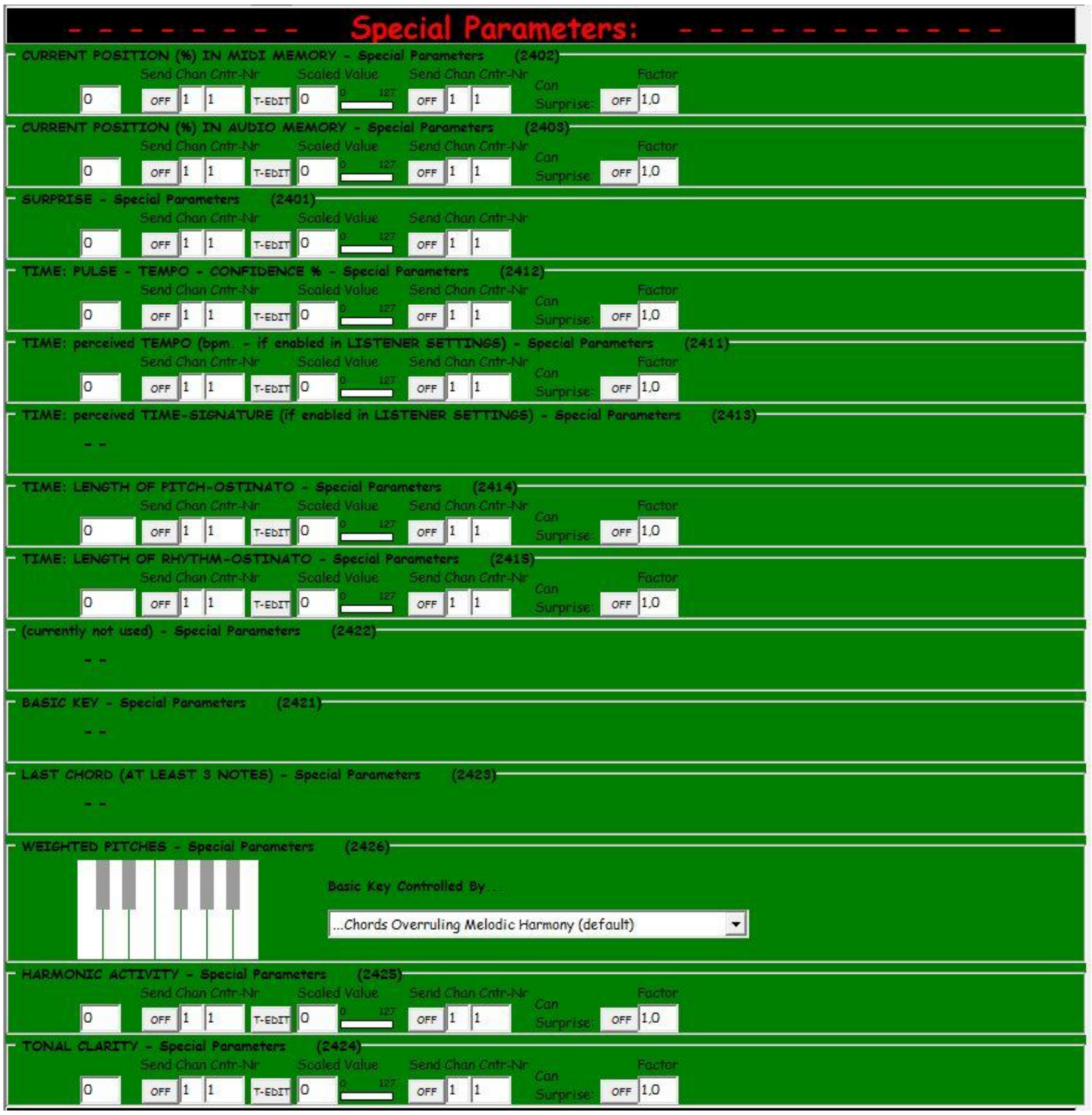
Average Can Surprise: OFF
Factor: 1,0
Scld. V.-Diffusion Send Chan Cntr-Nr: (Max-Min): 1 OFF 1 1

NOTE ACTIVITY in MEL.+ CHORDS - Aux Evaluation (S460)

		Send	Chan	Cntr-Nr		Scaled Value		Send	Chan	Cntr-Nr	Can Surprise:	Factor
Notes per Second:	0	OFF	1	1	T-EDIT	0	0 127	OFF	1	1	OFF	1,0

Special Parameters

This area contains a summary of parameters that work outside of the Main- and Aux.-Evaluations scheme and also don't concern the Last Events.


Special Parameters:

CURRENT POSITION (%) IN MIDI MEMORY - Special Parameters (2402)
0 | Send Chan Cntr-Nr: OFF 1 1 | T-EDIT | Scaled Value: 0 (0–127) | Send Chan Cntr-Nr: OFF 1 1 | Can Surprise: OFF | Factor: 1,0

CURRENT POSITION (%) IN AUDIO MEMORY - Special Parameters (2403)
0 | Send Chan Cntr-Nr: OFF 1 1 | T-EDIT | Scaled Value: 0 (0–127) | Send Chan Cntr-Nr: OFF 1 1 | Can Surprise: OFF | Factor: 1,0

SURPRISE - Special Parameters (2401)
0 | Send Chan Cntr-Nr: OFF 1 1 | T-EDIT | Scaled Value: 0 (0–127) | Send Chan Cntr-Nr: OFF 1 1

TIME: PULSE - TEMPO - CONFIDENCE % - Special Parameters (2412)
0 | Send Chan Cntr-Nr: OFF 1 1 | T-EDIT | Scaled Value: 0 (0–127) | Send Chan Cntr-Nr: OFF 1 1 | Can Surprise: OFF | Factor: 1,0

TIME: perceived TEMPO (bpm. - if enabled in LISTENER SETTINGS) - Special Parameters (2411)
0 | Send Chan Cntr-Nr: OFF 1 1 | T-EDIT | Scaled Value: 0 (0–127) | Send Chan Cntr-Nr: OFF 1 1 | Can Surprise: OFF | Factor: 1,0

TIME: perceived TIME-SIGNATURE (if enabled in LISTENER SETTINGS) - Special Parameters (2413)
- -

TIME: LENGTH OF PITCH-OSTINATO - Special Parameters (2414)
0 | Send Chan Cntr-Nr: OFF 1 1 | T-EDIT | Scaled Value: 0 (0–127) | Send Chan Cntr-Nr: OFF 1 1 | Can Surprise: OFF | Factor: 1,0

TIME: LENGTH OF RHYTHM-OSTINATO - Special Parameters (2415)
0 | Send Chan Cntr-Nr: OFF 1 1 | T-EDIT | Scaled Value: 0 (0–127) | Send Chan Cntr-Nr: OFF 1 1 | Can Surprise: OFF | Factor: 1,0

(currently not used) - Special Parameters (2422)
- -

BASIC KEY - Special Parameters (2421)
- -

LAST CHORD (AT LEAST 3 NOTES) - Special Parameters (2423)
- -

WEIGHTED PITCHES - Special Parameters (2426)
Basic Key Controlled By...
...Chords Overruling Melodic Harmony (default)

HARMONIC ACTIVITY - Special Parameters (2425)
0 | Send Chan Cntr-Nr: OFF 1 1 | T-EDIT | Scaled Value: 0 (0–127) | Send Chan Cntr-Nr: OFF 1 1 | Can Surprise: OFF | Factor: 1,0

TONAL CLARITY - Special Parameters (2424)
0 | Send Chan Cntr-Nr: OFF 1 1 | T-EDIT | Scaled Value: 0 (0–127) | Send Chan Cntr-Nr: OFF 1 1 | Can Surprise: OFF | Factor: 1,0

Current Position (%) in Midi/Audio Memory:
Both of these parameters give information as to the current place in which the two looping memories for Midi and audio can be found. If the raw value (to the far left) shows "0," either nothing is yet written into the memory, or it has just recoiled back to the beginning. The information comes in form of percentages, because you can specify the actual amount in the Listener settings under "Memory Size." This is practical if for example, you want to use several Listeners, some of which are only for the Midi output of a Player. They can then work as "critics" for Tango's own music. You can then reduce the (memory-gobbling) audio memory to a symbolic value of 1000 samples.

Surprise I have already described above. This parameter gathers the surprises which are evoked by the Evaluation parameters on which "Can Surprise" is activated.

> *You can specify its speed of reaction in the Listener settings under Surprise-Attack/Decay. How strongly Surprise reacts is regulated from the "Factor" settings of individual parameters and the amount of "surprise-active" parameters.*

Following are the parameters that have to do with time patterns such as

Tempi, Meter and Loops

Sometimes when you improvise you let your play flow in a "rubato" way without adhering to a particular tempo. At some point your foot might begin to tap and you may begin to associate every tone somehow with the tempo of your foot.

T² has a few sensors for that.

Time: Pulse Tempo Sensor that searches for signs if and in what tempo your "foot is tapping" and then shows the results of the analysis in two parameters:

Time: Pulse-Tempo-Confidence % (2412) shows how sure the Listener is that the current material is related to a tempo and

Time: perceived Tempo (bpm.) that, if applicable, displays the tempo in beats per minute.

If it hears a tempo, the Listener can then activate a Metronome if the following conditions are fulfilled:

- *It must be allowed in the **Listener settings** (in the upper right) to turn Tempo and Sync on and off – **Pulse Tpo. Sens.** must show "YES" for ON and/or OFF.*

- A Metronome must be connected to the Listener in the Room display via a black wire (Listener Output "Metronome" to the Metronome Input "Control").

- The raw confidence-value to the left must climb above 50%. That will only happen when you play a tempo rather precisely. Regardless of whether you play eighths, quarters or other note values, T² must be convinced that the movements of your fingers contain an inner pulse upon which it can centre itself.

With this configuration, if a tempo is detected you should hear the Metronome clicking at the correct speed.

Once the tempo is running, you can also play triplets or chains of dotted notes without the tempo getting lost – as long as you play them in time exactly enough.

In ListenerMetronomePulse.room, you can find out more about the mechanism.

> *In the **Listener settings** you have the possibility to specify how easily the threshold of 50% is reached with **Tempo 'ON' – Resistance**. T² is not so picky when this*

parameter is set to low values and it also accepts inexact playing. On the other hand, the program might hear material not meant to set a tempo as tempo-related.

You can also turn off a tempo in this way; when you go back to playing rubato, the Listener no longer hears the steady tempo from you and therefore shuts it off.

*You can set the readiness to turn off a detected tempo with the threshold **Tempo 'OFF' – Readiness** in the **Listener settings**. The value "0" would mean the same as the right **Pulse Tpo. Sens.** button in the "NO" position – with this setting no tempo can be switched off by the Listener.*

*The "Pulse Tempo Sensor" works with two very short **Evaluation times** (memory trails) that you can set in the **Listener settings**. Maybe with some testing you can find values more suitable for your needs.*

Apart from the "Pulse Tempo Sensor" there are two more tempo sensors which can detect a tempo and even the meter as well:

Time: Length of Pitch-Ostinato and
Time: Length of Rhythm-Ostinato.

These functions search for loops in your playing – that is to say, for material that is repeated several times. Often the loop is already found after the first repeat (the second time it is played); sometimes it takes a little longer. You can also help the Listener here if you play exactly.

Two types – rhythmic and melodic loops – can be heard.

If the pitch pattern is not repeated but there is a rhythmical Ostinato in your playing a loop will be detected by the

"**Length of Rhythm-Ostinato**" (Parameter 2415), as long as you repeat the rhythm.

In contrast, if there is no perceivable rhythmic pattern, e.g. running eighths (played in time), in which only a pitch sequence is repeated, then the

"**Length of Pitch-Ostinato**" (Parameter 2414) reacts.

Both parameters give you the length of the loop in seconds, the tempo and the meter.
They can also turn a Metronome-tempo on or off, if

- *"Pitch Loop Sens." and "Rhythm. Loop Sens." are allowed to turn on Tempos in the upper right of the **Listener settings**,*
- a Metronome is connected to the Listener via a black wire in the Room view (Listener-output "Metronome" to the Metronome–input "Control") and if
- The Listener effectually hears a loop.

Note: Allow only the Pulse Tempo Sensor **or** the two loop sensors to turn on the Metronome-Tempo, or there may be very confusing results. ("Where in the world did that tempo come from?!") The loop sensors do not get in each other's way.

> *If you also allow a meter to be set in the upper right of the* ***Listener settings (Set Time Sgn.)****, the loop sensors will also hear that meter and simultaneously redirect the Metronome. These are displayed in*

Time: perceived Time-Signature. Of course you can see this in the Metronome in the Room view as well.

> *In* ***the Listener settings*** *you can also choose the settings for the* ***Maximal Loop Length****. The loop sensor cannot hear loops longer than those set here.*

You can find an example for the loop sensor in ListenerMetronomLoop.room.

Harmonic Listener Parameters

Basic Key looks in the Listener input for the basic harmonic area through which the music is currently passing.

This parameter considers both melodies and chords. For example, if F major is the “Basic Key” being played, while in the **melody lines** d minor, a minor, g minor or the dominant seventh C7 are quickly touched on, the “Basic Key” in the display will remain “F-Maj.” If you modulate to d minor however, not just wandering briefly through the key but perhaps emphasizing a C# as a leading tone, or lingering in d minor a little longer, the Basic Key will follow you.

With this algorithm, I hope to have reached a good balance between the necessary inertia when listening to diatonic material and a sufficient tracking speed (as shown, for example, with the leading tones of minor keys).

Last Chord analyses chords in your playing and also sets the Basic Key. Chords are only analysed, if they consist of at least **3 notes.** So overlapping piano-notes, octaves etc. don’t appear in this analysis. Dominants are not passed on to the above parameter as “Basic Key” but as its dominants, so a G7 chord will still appear as a C major or minor, depending on the context.

Weighted Pitches is the central parameter of the Listener’s harmonic system. All harmonic Listener parameters use Weighted Pitches for their work.

It keeps track of the **harmonic significance** of the last notes you just played: For that it does not only look at the pitches but also at the volumes, the “age” and (for melodic material, not for chords) also at the length of the notes.

Here the model is derived from the way an improviser would perceive the harmony of a human partner:

- older material gets quickly forgotten,
- soft and/or short grace notes are not as important as loud and long ones and
- the pitch register is not as important as the pitch class (C, C#, D etc.).

With chords there is obviously more definition, less fuzziness than with melodic material: On the right you can decide, if

- chords overrule melodic findings,
- are equally treated as melodic harmony,
- melody comes “first” or
- harmonic content of melodies is completely ignored for the Basic Key.

Harmonic Activity (2425) and
Tonal Clarity (2424) can be converted to Midi Controllers and led out of the Listener.

Harmonic Activity is a type of surprise function for harmonic changes. A change from F major to Bb major contains much less harmonic activity than a change from F major to Db minor simply because the former two keys have a much closer tonal relationship to one another.

Tonal Clarity looks at how strongly the Basic Key is supported by the notes used: root and fifth of the Basic Key are very clear, whereas the major third in a minor key lowers the clarity quite a bit – as, overall, chromatic alterations lead to a further loss of tonal clarity.

For more on harmony and "Harmony", the module that encapsulates the harmonic abilities of T² and its collaboration with the harmonic Listener-parameters, see the chapter on Harmony in this manual. Harmony will probably be completed by the end of 2015.

The Parameters for “Last Events”

describe what has happened last or what is happening right now.

Current Rest: The length of the rest since the last note you played. If you are right in the middle of playing a note of course, it is “0.”

Last Rest on the other hand, notices the last rest even when in the middle of playing a note. Also, while you are in the middle of a rest, “Last Rest” shows the previous rest (the rest preceding the rest you are now taking) but only until the current rest is longer than the previous rest. From then on, “Current” and “Last Rest” have the same value, a value which grows the longer you don’t play.

Duration of Current Input Activity: Given that you are playing right now – how long have you played since your last rest? With this parameter, you can allow Player modules to interrupt you when you forget your counterpart (Tango) and play your own “solo” without listening to the software anymore. Alternatively, a longer phrase from you causes T^2 to adapt to your harmony or your tempo in a spot where the program would normally be allowed to act more independently – these are only two of many possibilities how to use this parameter in a Room configuration. Many such functions are configured in the examples ErstesKennenlernen.room and TwoDuos.room. More information is available in those Rooms’ “Info”.

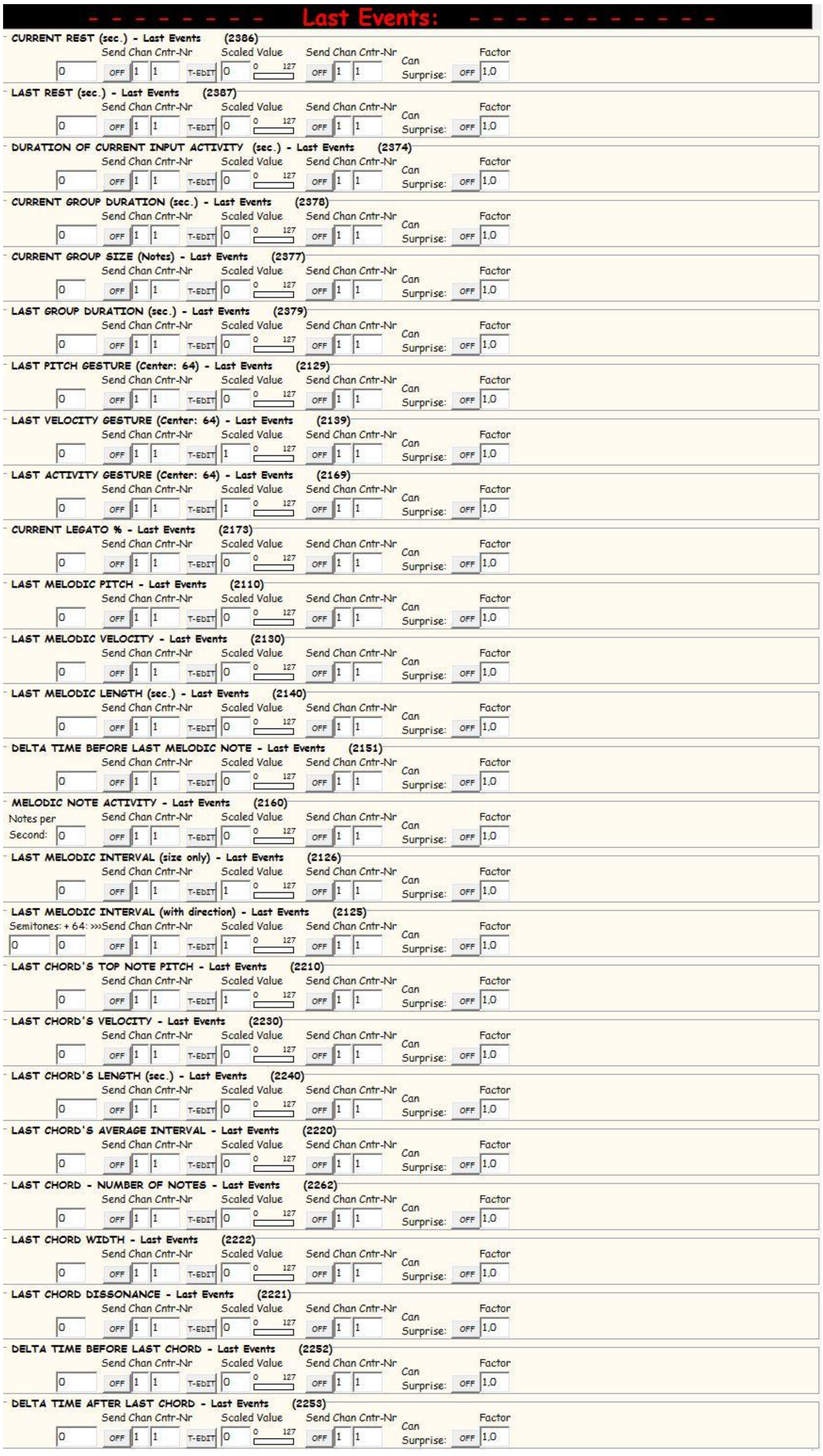
Current Group Duration is very similar, not resetting to zero after you cease playing, but continues to display the length of the last phrase. If you play a new phrase, this parameter, just like “Duration of Current Input Activity,” goes back to zero and shows the increasing length of the ongoing phrase.

The length of your last phrase often has a huge effect on a human duo partner: the end of a long phrase means much more and is also often more theatrical than something which is quickly thrown out there.

Current Group Size tells you how many notes the last Group contained.

> *You can set what the Listener’s definition of a Group in the **Listener settings** by **Min. Rest before a new Group**. Most of the time a Group follows a rest and its minimum length is set here. If you would also like a long note (as the closing note of a phrase) to be able to conclude a Group, then you can define its minimum length under the parameter Min. **Note Length before new Group**.*

Last Group Duration relates to “Current Group Duration” the same way “Last Rest” does with “Current Rest.” It shows the last **finished** Group except when the current Group is longer than previous one. This parameter avoids a phrase length of 0.1 seconds being re-set with every phrase which begins because you simply began a new phrase. Thus it prevents a re-set to zero at the beginning of a phrase.


- - - - - - - - - Last Events: - - - - - - - - - - - -

CURRENT REST (sec.) - Last Events (2386)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

LAST REST (sec.) - Last Events (2387)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

DURATION OF CURRENT INPUT ACTIVITY (sec.) - Last Events (2374)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

CURRENT GROUP DURATION (sec.) - Last Events (2378)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

CURRENT GROUP SIZE (Notes) - Last Events (2377)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

LAST GROUP DURATION (sec.) - Last Events (2379)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

LAST PITCH GESTURE (Center: 64) - Last Events (2129)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

LAST VELOCITY GESTURE (Center: 64) - Last Events (2139)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 1 0 127 OFF 1 1 OFF 1,0

LAST ACTIVITY GESTURE (Center: 64) - Last Events (2169)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 1 0 127 OFF 1 1 OFF 1,0

CURRENT LEGATO % - Last Events (2173)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

LAST MELODIC PITCH - Last Events (2110)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

LAST MELODIC VELOCITY - Last Events (2130)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

LAST MELODIC LENGTH (sec.) - Last Events (2140)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

DELTA TIME BEFORE LAST MELODIC NOTE - Last Events (2151)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

MELODIC NOTE ACTIVITY - Last Events (2160)
Notes per Second: — Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

LAST MELODIC INTERVAL (size only) - Last Events (2126)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 1 0 127 OFF 1 1 OFF 1,0

LAST MELODIC INTERVAL (with direction) - Last Events (2125)
Semitones: + 64: >>>Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 0 OFF 1 1 T-EDIT 1 0 127 OFF 1 1 OFF 1,0

LAST CHORD'S TOP NOTE PITCH - Last Events (2210)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 1 0 127 OFF 1 1 OFF 1,0

LAST CHORD'S VELOCITY - Last Events (2230)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

LAST CHORD'S LENGTH (sec.) - Last Events (2240)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

LAST CHORD'S AVERAGE INTERVAL - Last Events (2220)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

LAST CHORD - NUMBER OF NOTES - Last Events (2262)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

LAST CHORD WIDTH - Last Events (2222)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

LAST CHORD DISSONANCE - Last Events (2221)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

DELTA TIME BEFORE LAST CHORD - Last Events (2252)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

DELTA TIME AFTER LAST CHORD - Last Events (2253)
Send Chan Cntr-Nr — Scaled Value — Send Chan Cntr-Nr — Can Surprise: — Factor
0 OFF 1 1 T-EDIT 0 0 127 OFF 1 1 OFF 1,0

Last Pitch Gesture,
Last Velocity Gesture,
Last Activity Gesture:
Gestures are short movements that you make to generate an effect. The Gesture parameters consider the last Groups of notes (explanations about Group settings can be found two paragraphs above this one), or your last phrase. The middle value (64) means there is no gesture to be found in this area. Values over 64 indicate an ascending tendency, values under 64, a falling tendency. This effect can be compared for example, to a conductor who pulls the music along by lifting his hands and palms, asking for more volume. This parameter is available for pitch, volume and note density.

Current Legato: How big are the current intervals of time between consecutive notes? Not just the pause is measured; instead the relationship between note lengths and the following rest is considered. 50% means that the note and rest are just as long as each other.

Last Melodic Pitch,
Last Melodic Velocity,
Last Melodic Length:
Here you find the values of the last-played note. By "Last Melodic Length" the value of the current sounding note is displayed and actualized as soon as it is longer than the previous note.

Melodic Note Activity: The average note density in the last two seconds.

Last Melodic Interval (size only): The size (independent of direction) of the interval between the last melody note and the note before that.

Last Melodic Interval (with direction): Here, **next to the size, the direction** is also specified. When no pitch change takes place between the two consecutive notes the raw value is 64, regardless of the register (region on the keyboard) you are playing in. A minor third downward leads to the value 61 (64-3 half steps).

The specification of the last chord:

Last Chord's Top Note Pitch and

Last Chord's Velocity specifies the pitch and volume of the last chord's highest note.

Last Chord's Length: Here the length of the currently-sounding chord is displayed and actualized as soon as this length is longer as the one before it.

Last Chord's Average Interval: Only the intervals between chord notes sorted by pitch are considered.

Last Chord's Number of Notes and
Last Chord's Width specifies the size of the last chord according to the number of notes and the interval between the highest and lowest notes.

Last Chord's Dissonance is not dependent on the number of notes but rather on the interval structure within a chord. Sevenths and seconds are more dissonant than octaves, fifths or thirds.

Delta Time before Last Chord: Often the meaning of a chord defines itself through the fact that before it, no other chord has been played for some time.

Delta Time after Last Chord: The time passed since the end of the last chord you played.

> ***About the Definition of "Chord":*** *In the Listener settings as already described, you can specify how large the maximal interval of time between chord notes can be (Max. Delta-Time between Chord-Notes – 40 milliseconds is the preset default). All notes that are further apart do not belong to chords.*

This concludes the manual of Tango², Version 1.8.5

Viel Spaß!

Köln, January 2015

Henning Berg

Appendix

The Installation of Windows7 and Tango² on the Mac with Boot Camp

Computer used:	Mac Book Pro
Processor:	2.26 GHz Intel Core 2 Duo
Mac OS:	OS X, V.10.9.4

Windows operating system: Windows 7

Tango² version 1.842

• Download Windows7 from the internet. There are free trials, which must be converted after 30 days or so into full versions.
.

• Use the Disk Utility (installed on the Mac) to burn the downloaded .iso file to a DVD.

• Run Boot Camp Assistant. Create a USB stick (about 5 GB) on which the wizard stores the automatically downloaded drivers that are required for Windows.

• Define the partitions using the wizard: At least 25 GB – it can be a bit more - must be reserved for Windows.

• Restart the Mac. Immediately after the welcome message appears, press "Alt" and select Boot Camp.

• Windows7 Installation Wizard: Select Boot Camp> Advanced Settings> Format the Partition Boot Camp, then install the Windows 7 DVD in the partition Boot Camp. Then start Windows.

• On the USB stick, look for the file Setup.exe and start it. This will install the drivers which are necessary for Windows.
The drivers for the sound card (needed for audio software) are as yet not present and must be downloaded and installed separately. You should in any case use a decent external sound card that runs under ASIO for more fun with T².
ASIO is the standard for the audio handling on the PC and is usually not supported by the built-in sound systems. ASIO also provides Midi handling.

• From here, everything looks and works like on any Windows PC.

• Download and unzip Tango² from www.henning-berg.de. There is no installation routine. Simply place the unpacked folder on the desktop and leave the file structure as it is.

• When I installed an M Audio card there was a problem with the message "*The ASIO sample rate is not supported. Please check your sync settings in the Control Panel.* "
I had to go to the Windows Control Panel and declare the built-in sound card for Mac the default device. After that M Audio worked without problems.

• Note: After starting Tango² in the menu "Extras>Global Audio and Midi Settings" the "Midi In Device" and the hardware for audio in and out have to be set. This is automatically saved, when you shut down T².
Additionally, with the first opening of every example-Room go to the module Audio / Midi Out (mostly on the right of the Rooms) to define the Midi output of your system. Save the Rooms with this setting, if you had to make changes here.

• Windows at some point will automatically try to update to its latest version (at the very latest when you turn your computer off). You should allow this. It may take 1 hour.
Do not turn off the computer during the process.