

# Tango<sup>2</sup>: Software for Computer-Human Improvisation

Henning Berg  
Försterstr. 25  
50825 Köln, Germany  
www.henning-berg.de  
Professor for Jazz Trombone  
Hochschule für Musik und Tanz Köln

## ABSTRACT

This paper describes **Tango<sup>2</sup>**, software for Computer-Human Improvisation developed for more than 25 years by Henning Berg.

Tango<sup>2</sup> (T<sup>2</sup>) listens to an improvising musician, analyses what it hears and plays musical responses which relate directly to the musical input.

If the improviser in turn reacts to these answers, a musical loop between the human and the machine can emerge. The way input and reaction correlate and the predictability of Tango's responses can be defined by the user via a setup of improvising environments, called Rooms.

Real-time sampling with knowledge of the musical content behind the samples and Midi-handling are unified via Tango's own monophonic audio-to-Midi, time stretching and pitch shifting algorithms. Both audio and Midi can be used by Tango's modules (e.g. Listeners, Players, Modifiers, Metronomes or Harmony) for input and output.

A flexible real time control system allows for internal and external remote control and scaling of most parameters.

The free software for Windows7 with all necessary folders, English and German manuals, many example-Rooms and a few videos can be downloaded at [www.henning-berg.de](http://www.henning-berg.de).

## Author Keywords

improvisation software, computer-human interaction, feedback loop

## 1. TANGO FOR THE ATARI AND TANGO<sup>2</sup>

After hearing George Lewis perform with his Voyager at the ICMC 1988 in Cologne, I decided to learn how to program in C and to develop an interactive system for myself. Improvising with a machine was never meant to be a replacement or a practice tool for playing jazz with real people. Instead, it requires a rather different approach and has a quality of its own.

During the nineties I played many gigs with my first improvising software Tango for the Atari, published by Steinberg 1991. The band "Tango & Company" with the late English jazz-pianist John Taylor, Tango and myself performed at important festivals in England (Bath) and Germany (e.g. Berlin), played gigs for West German Radio (WDR) and recorded a "Trio"- CD („Tango & Company“ 1997, Jazzhaus Musik).

Tango for the Atari was the first program I had written, so due to its limited architecture and many additions, it started to lose some of its former stability and finally also its hardware platform, the Atari. Thus, in 1998 it was time to start over.

With Tango<sup>2</sup> I try to incorporate my concert- and programming experience in a software that addresses many of the old, basic ideas in a more flexible and comprehensive way.

## 2. BASIC IDEAS

The key feature of Tango<sup>2</sup> is its clear modular design: There are many different software devices (modules), which the user can use and connect in a Room. There are Listener-, Player-, Modifier- and other modules. They can be applied very flexibly to accommodate for simple or complex stimulus-response configurations according to one's needs.

Particularly important is the integration of Midi and audio in all parts of the program: T<sup>2</sup> has a built-in audio-to-Midi module for monophonic instruments such as wind-instruments, voice and strings. This makes it possible to communicate with the program via a Midi keyboard or directly via a microphone, without an external pitch-to-Midi device. Additionally, T<sup>2</sup> can not only play Midi sound generators but also use and modify the audio-signal it received moments ago from the human partner for its responses. The same modules are used for audio and Midi functions so it makes no difference if Tango<sup>2</sup> works with Midi or with audio. This is possible, because T<sup>2</sup> analyses the musical structures behind the samples as if they were Midi signals.

The user configures Rooms, i.e. environments, in which he will later improvise. A concert with Tango<sup>2</sup> usually consists of one "Room" or a "Suite" of several rooms, which are called upon in order to improvisationally pass through or move within.

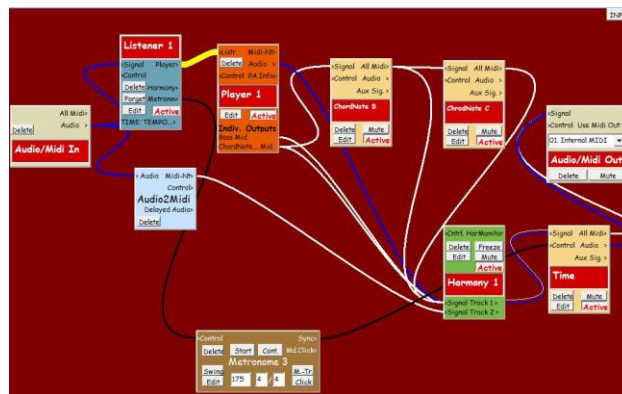


Figure 1. A very simple Tango<sup>2</sup> Room. The video for this demo was produced using this Room.



Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). Copyright remains with the author(s).

### 3. MODULES

Many modules of the same or different types can be part of the same Room. For example, one Listener can be used for the user's input, but another Listener can at the same time listen to the output of a Player in this Room. This way T<sup>2</sup> can also control and adjust its own output. Important modules are:

#### 3.1 Listener

The Listener remembers the most recent music it received over its signal input – note for note. The user can determine a memory loop for audio and Midi data, therefore a trail of data for the Evaluation(s) is always available. If audio (converted to Midi by the Listener's audio to Midi) 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, Tango's audio memory, other than 'normal' sampling memories, knows about the musical content of what is saved.

The input is then analysed with more than 200 parameters. These function on different time scales (three short/long-term memories) and musical topics (e.g. intervals, harmony, tempo, loops, density, surprise...).

The results of this "Evaluation" can be scaled and wired for real time control of other parameters during a performance.

#### 3.2 Player

The Player uses the Listener's memory to generate Tango's responses. It can operate one or more tracks that play either simultaneously or alternately. With these responses the Player can react and also act in a rather flexible and, if that is desired, surprising way.

Each track alone can produce monophonic or polyphonic material, including chords (Figure 2).

The Player's output can then be sent to other modules to modify it according to the current needs.

#### 3.3 Modifier

The Modifier alters properties of Midi- or audio-notes (e.g. intervals, volume, legato-quality, quantization). The degree of those changes can be tied to defined conditions and – again – modulated via real time control.

### 3.4 Harmony

The module's purpose is to bend music, which is played by a Player without harmonic "objectives", so that the result will sound harmonically organized.

This can be done via a Leadsheet (which is common in Jazz), or controlling and modifying the dissonances, as they (happen to) come from a Player.

Furthermore there is the possibility to develop individual systems of chords and scales, defining the "Harmonic Relevance" of each scale note with T-shirt sizes from XS to XL.

Harmony can also be controlled by the Listener's harmonic Evaluation or come up with its own variations of given tonalities, e.g. "for harmonic variation use 20% diatonic steps with secondary dominants and only 5% diatonic modulations".

Other module types are a Metronome and, of course, input/output modules.

### 4. FURTHER WORK

Tango<sup>2</sup> is a work in progress. As of April 2016 I am working at a Listener function to get a percentage of confidence, whether my playing has to do with a certain melody – even, when it is played in a rather abstract way with many added or left out notes, different key or tempo or not from the beginning.

Next will be modules that organize time and rhythm in a more subtle way, something that already existed in the old Tango for the Atari, but had no priority until now for T<sup>2</sup>.

Also, the GUI needs some work, e.g. groups of modules should work hierarchically to be able to simplify Rooms.

Finally I want to work at modules that play modules rather than music, i.e. work on a higher level.

But most of all, I am glad to be able to play concerts with Tango<sup>2</sup> again after 18 years of developing it.

### 5. ACKNOWLEDGEMENTS

Thanks to Clarence Barlow, Werner Kracht, George Lewis Robert Rowe and Christoph Windheuser for many good ideas and great inspiration.

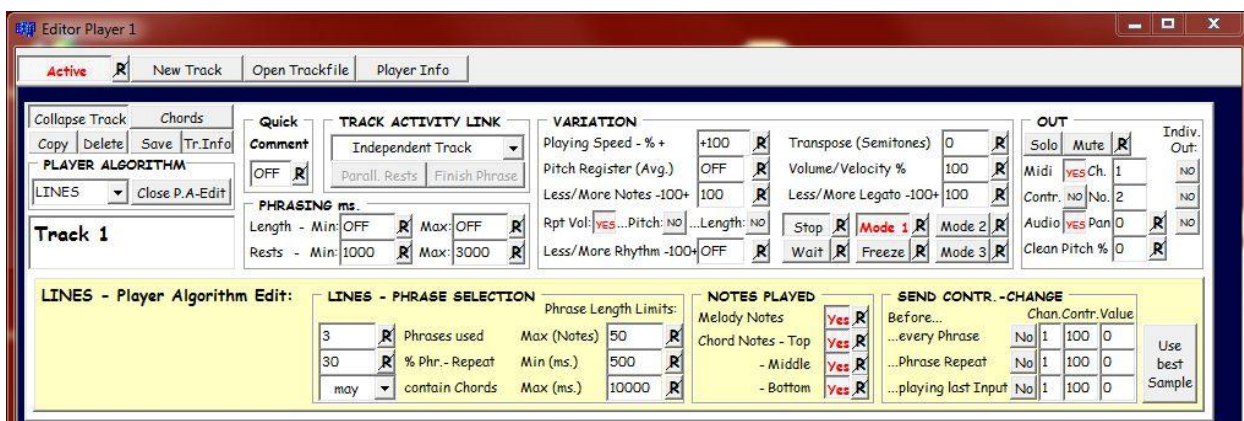


Figure 2. The Player Editor with one track.