

Sound Design Toolkit (SDT)

A Toolkit for Exploration in Sonic Interaction Design

1. Introduction

The Sound Design Toolkit (SDT) is a software package consisting in a set of physics-based sound synthesis models, available as a complete front-end application for Max/MSP/Jitter. In particular, the package includes polyphonic features and connectivity to multiple external devices and sensors in order to facilitate the embedding of sonic attributes in interactive artifacts [1].

The available palette of physics-based virtual lutheries and foley pits, developed and implemented as externals and patches, can be easily coupled with physical objects and are computationally affordable for real-time applications on ordinary hardware.

The sound models algorithms are developed according to three main points: 1) auditory perceptual relevance; 2) *cartoonification*, i.e. simplification of the underlying physics and emphasizing of its most relevant aspects in order to increase both computational efficiency and perceptual sharpness; 3) parametric temporal control ensuring appropriate, natural and expressive articulations of sound event sequences (for an in depth description of the sound models please refer to the CLOSED Deliverable D2.3 - Sound Design Toolkit Instruction's Guide [2]).

The main features of the current release of the SDT are:

- a high level, general template of available sound models, organized according to a taxonomy of everyday sounds;
- the clustering of control parameters into high and low level GUIs according to their more or less meaningful and immediate effects, thus providing a clear monitoring of the sound models;
- an intuitive understanding of the control parameters, facilitated by a common sense and perceptually-based naming and ranging of the physical parameters;
- the possibility to allocate multiple instances (polyphony) of sound models in order to facilitate the design of compound sound events;
- a comfortable integration of external devices allowing an interactive control of the sound models
- the possibility to save and recall complex arrangements of multiple sound models and their settings.

2. Installation

1) In "SDT_externals/" there are 2 zip files respectively containing the externals for mac and win, and a sub-directory "SDT_common_abstractions/" which contains all the common abstractions used in the SDT.

1A) Copy the whole directory "SDT_externals/" to "Max6/Cycling '74/";

1B) Get inside the copied directory and delete the zip file which is not for your OS (e.g., delete the win-related file if you are on a mac);

1C) unzip the compressed file for your OS inside "SDT_externals/";

2) Copy the directory "SDT_help/" to "Max6/Cycling '74/";

3) Copy the file SDT-objectlist.txt from "SDT_init/" to "Max6/Cycling '74/init/";

4) Copy the file SDT_overview.maxpat from "SDT_extras/" to "Max6/patches/extras/".

Usage: Get inside "SDT_patches/" and open SoundDesignToolkit.maxpat

Note: The SDT makes use of a few third party externals, which are listed in the included files credits_3rd-party_mac.txt and credits_3rdparty_win.txt

3. Navigating the SDT

The SDT is composed of a modular framework of patches and sub-patches, arranged in folders and subfolders, as shown in Figure 1.

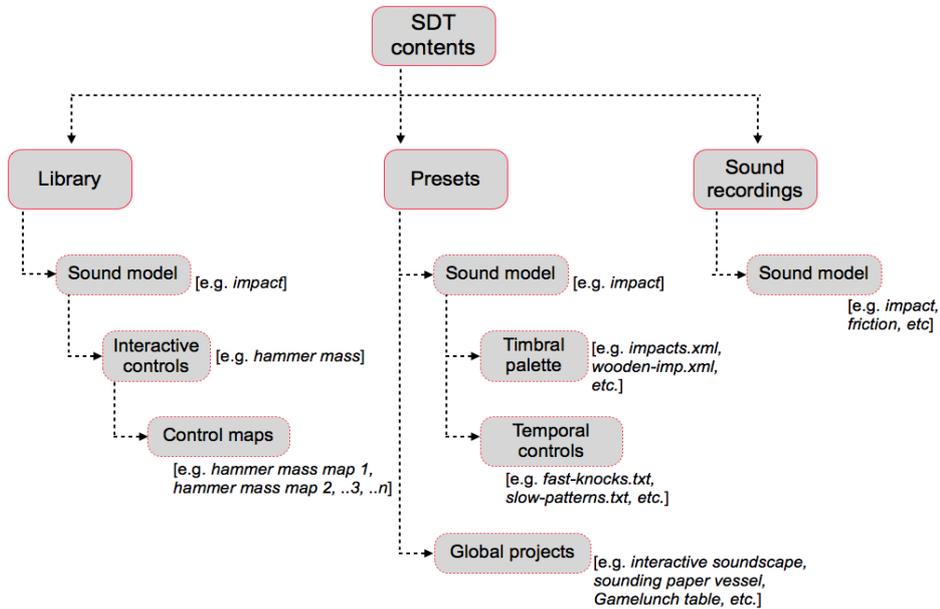


Fig. 1: The SDT tree of folders and sub-folders.

In the *Library* folder, sub-folders contain all the patches and abstractions that compose the modular structure of the sound models. In the *Presets* folder, “temporal control” and “timbral palette” presets are saved as XML or TXT file in as many sub-folders as the sound models. In the *Sound recordings* folder, samples can be recorded as AIFF files.

Figure 2 shows the hierarchical arrangement of the available sound models, starting from low-level sound events to more complex, patterned or compound processes. The corresponding low-level sound models are presented at the bottom of the graph, while the second level shows basic events and sound textures straightly derivable from them. Processes that can be related to temporal patterns of basic events and textures are presented in the third level. The top level contains several examples of the implemented simulations, while dashed connections represent expected dependencies for the simulations yet to be developed. Therefore, in the SDT several sound models can be combined to compose complex, interactive compound events that can be arranged and saved as projects.

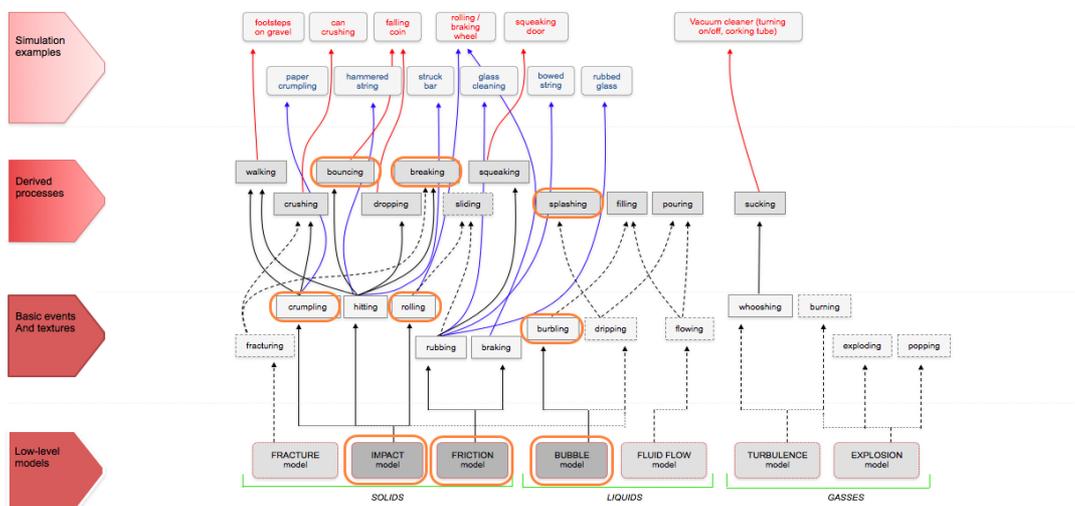


Fig. 2: A taxonomy of everyday sound models.

4. The SDT Front-end

To launch the SDT double-click on *SoundDesignToolkit.maxpat* patch. In Figure 3, a front-end shows up displaying the available sound models, hierarchically organized.

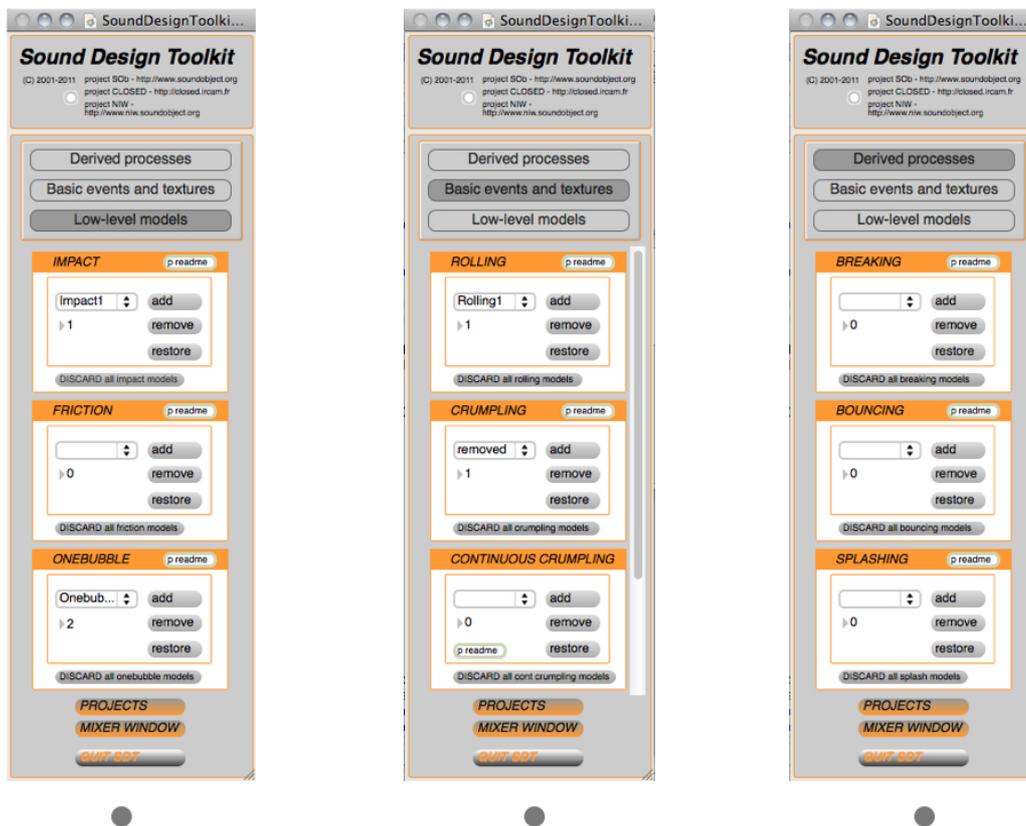


Fig. 3: SDT front-end, view of the navigation tabs, displaying the hierarchy of available sound models

4.1 The Sound Model Manager

The overall framework of the SDT allows to manage multiple instances of sound models within a single working session. In the sound model manager GUI shown in Figure 4, instances are added in ascending order and can be browsed, removed or restored, at user's choice.

add: loads and adds instances of the sound model. Added instances are displayed as items in ascending order in the drop-down menu.

remove: removes the current instance displayed in the drop-down menu. The corresponding item is displayed as "removed".

restore: "removed" items (instances) in the drop-down menu can be restored.

discard all XXX model: instantaneously closes all the loaded xxx instances.

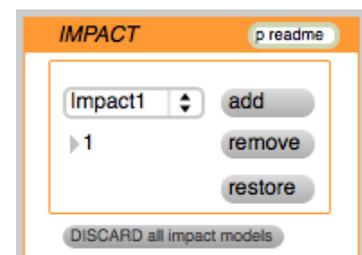


Fig. 4: detail of the sound model manager GUI.

4.2 The Mixer Window

A separate window, accessible in the front-end patch, provides the global audio mixer, shown in Figure 5. Volume, panning controls, audio sends and recording tools of currently loaded instances are automatically generated and displayed.

Recording a sound: click on "recorder" to access the recording tools. The single audio output can be recorded as stereo AIFF file. The "open" box opens a dialog window to create and locate the file on disk, while the

adjacent switch starts/stops the recording. The “sends” box shows a window with two automatically generated [send~] objects, in order to potentially apply further sound processing.

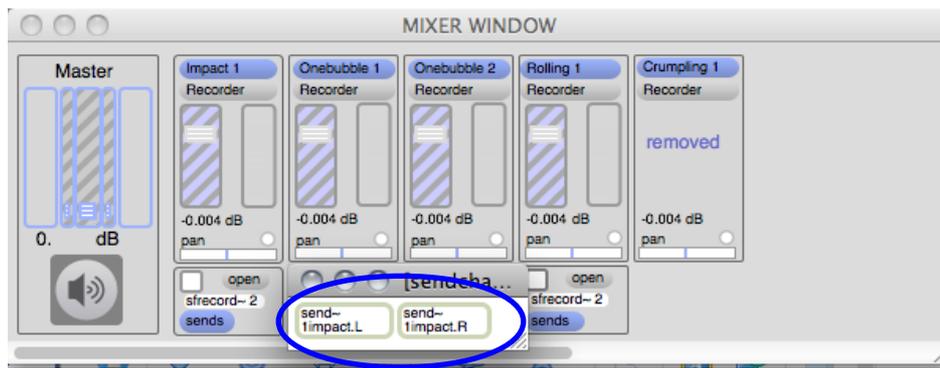


Fig. 5: In this example, the working session makes use of one impact, two bubble models, one rolling and one crumpling (removed). Audio sends for impact 1 are shown.

5. The Sound Model GUI

For each sound model, the available control parameters are assigned either to low- or high-level interfaces. The GUI provides a functional hierarchy of parameters: the most effective control parameters are displayed in the high level interface, though their accessibility is retained in the low-level interface too.

5.1 High Level GUI

The high-level interface is organized in two main sections, as shown in Figure 6:

- 1) the “**physical parameters**” encompass the most direct and effective parameters;
- 2) the “**timbral palette**” allows to manage, store, recall, delete or interpolate configurations of physical parameters. Configurations are saved as presets in XML files, thus allowing them to be read, modified or generated from other XML-compatible software.

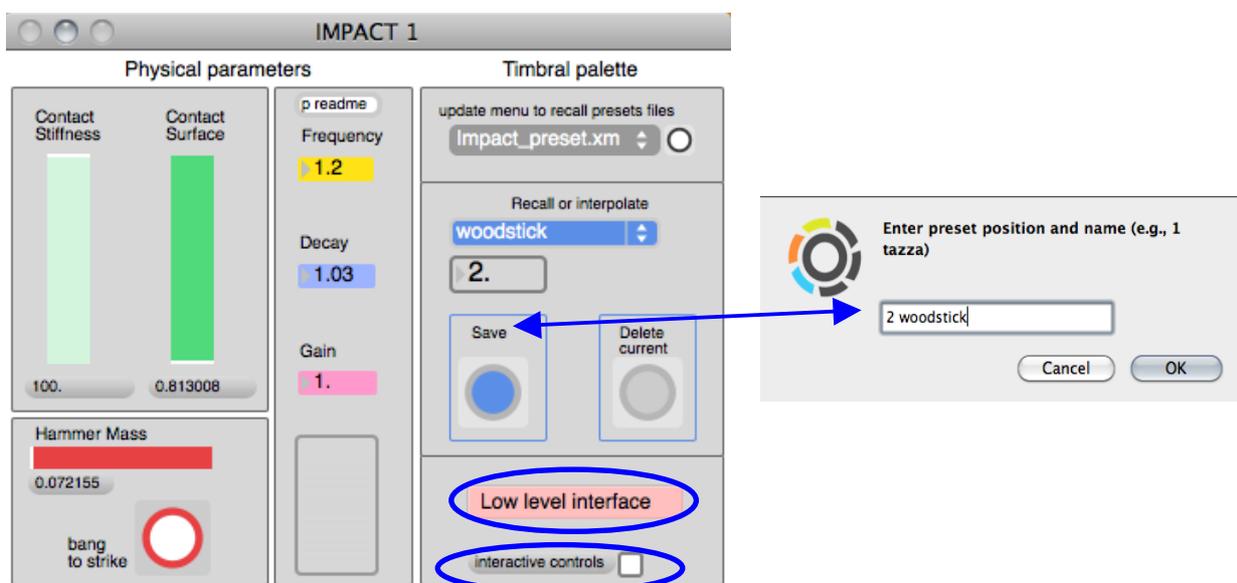


Fig. 6: High level GUI, detail of the dialog window to store configurations of parameters (index position relating to the blue drop-down menu and preset name).

PHYSICAL PARAMETERS: physical parameters displayed in the high level GUI are interactively controllable with external devices (see further *INTERACTIVE CONTROLS*). The *frequency*, *decay*, *gain* boxes work as global factors of parameters values that can be respectively accessed and manipulated in the low level interface.

TIMBRAL PALETTE: timbral preset.xml files are located and must be saved in `SDT_contents/Presets/[SOUNDMODEL]/Timbral`. When the current sound model instance is removed a dialogue box shows up asking to save the timbral preset file.

grey drop-down menu: shows the currently available preset.xml files. The adjacent white button populates the drop-down menu with the items (preset.xml files) currently present in the timbral folder.

blue drop-down menu: displays the timbral configurations (presets) available in the currently loaded preset.xml files.

blue button: opens a dialog window to store or replace new presets in the blue drop-down menu. Enter the new menu item position and preset name (spacing in preset names is not allowed, i.e. 3 metallic_coin).

grey button: deletes the current preset in the preset.xml file.

red box (low level interface): opens/closes the low level GUI.

grey box (interactive controls): gives access to the *interactive controls* section. This section provides sequencing tools for off-line simulations of gestures and patterns, and control patches of interactive parameters with external devices.

5.2 Low Level GUI

The low-level Interface, shown in Figure 7 gives access to all control parameters available in the model, for further tuning and refining the sound. For example, it is possible to activate, deactivate or mute each resonant mode and modify its frequency, decay time and gain.



Fig. 7: the low-level GUI of the impact model. The hammer controls refer to the physical parameters of the virtual striker, the resonator controls refer to the spectral characteristics (modes of resonance) of the struck object, the impact parameters refer to the quality of the contact between the two virtual object.

The GUI is framed according to the general modular structure of the sound models “resonator(object1)--in-teractor--resonator(object2)”, representing the interaction between two resonating objects, as shown in Figure 8.

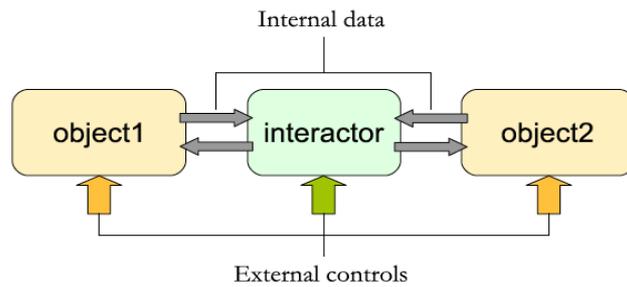


Fig. 8: the modular framework of the sound models, describing the interaction between two resonating objects.

For the in-depth description of all the sound models and their control parameters, please refer to the CLOSED Deliverable D3.2 - Sound Design Toolkit Instruction’s Guide (available at http://closed.ircam.fr/uploads/media/CLOSED_D2.3.pdf).

5.3 Interactive Controls

The *interactive controls* section is accessible through the high-level GUI. This section, shown in Figure 9, includes sequencing tools for the simulation of gesture and patterns, and control patches to interactively manipulate the physical parameters with external devices.



Fig. 9: The interactive controls section gives access to sequencing tools and interactive parameters control maps.

5.3.1 Connectivity to External Devices

All the parameters displayed in the high-level interface of each sound model are interactively controllable with external devices. A collection of 3rd-parties communication patches is available in the [Library/3rd-](#)

parties_MIDI-OSC_communication folder. The control parameters colored buttons give access to the control maps GUI (shown in Figure 10), where set of control map patches are stored and available for the user. Once established the MIDI/OSC connection with Max/MSP, it is sufficient to make use of [send/receive] objects directly in the control map patches. It is, indeed, possible to edit and save, as Max/MSP patch, each interactive parameter map. Newer maps are automatically listed and can be recalled in the drop-down menu. Such a feature makes it possible to collect a set of different solutions that can be easily combined at user's choice. In practice, the user has an auditory analogue of the board used by designer to rapidly compare a large number of drawn sketches.

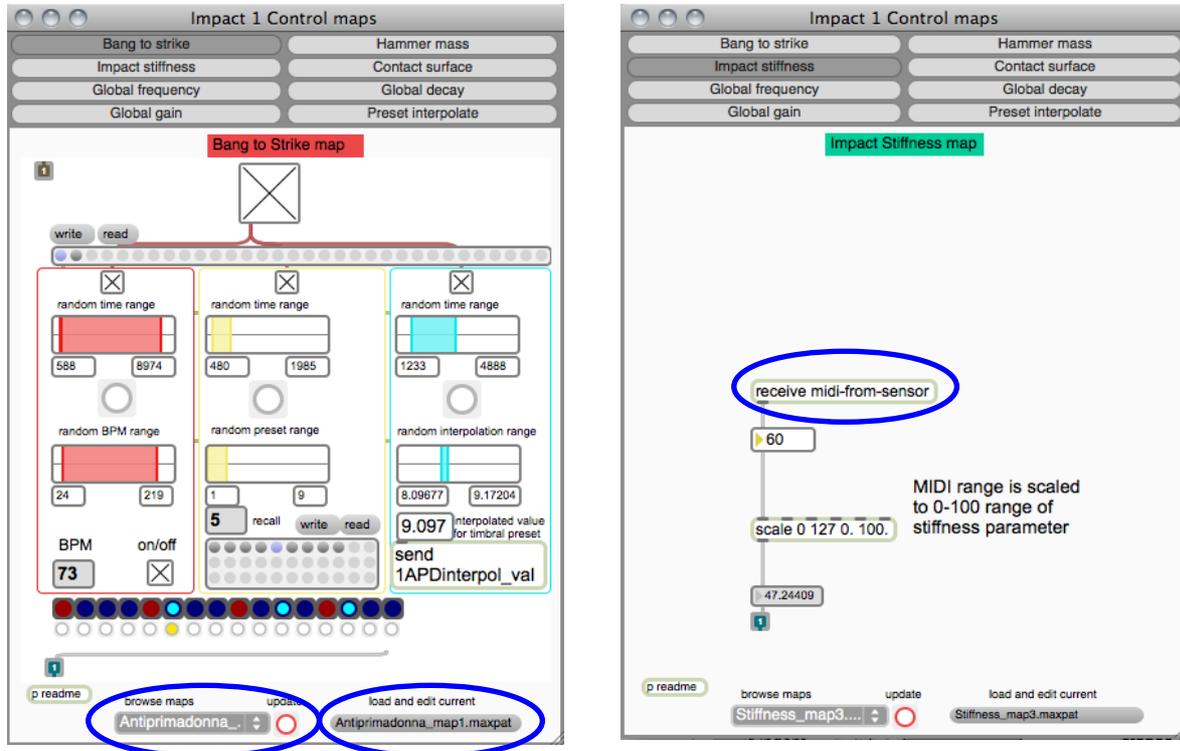


Fig. 10: Example of control maps GUI, parameters maps can be recalled from the drop-down menu, edited, saved and collected for prompt use.

parameters boxes (grey): allow tab navigation of parameters control windows.

drop-down menu (grey): displays the currently collected maps, as Max/MSP patches, that are located in the SDT_content/Library/[SOUNDMODEL]/MIDI_OSC_[SOUNDMODEL]/[PARAMETER] folder. When new maps are added to the folder, the adjacent white button allows to refresh the items listed in the menu.

grey box (load and edit current): displays the current map. It works as button to load and edit the Max/MSP patch.

Note: control parameters range is generally displayed in 0-100 floating. Incoming control signal must be scaled within that range in the map patches before the outlet which is connected to the parameter. Edited or new maps must be saved in SDT_content/Library/[SOUNDMODEL]/MIDI_OSC_[SOUNDMODEL]/[PARAMETER] folder.

6. SDT PROJECTS

In Figure 11, complex configurations of sound models¹ (sound models and number of instances, timbral presets, control maps used, audio mixing, etc.) can be stored and saved as global projects, in the SDT_contents/Presets/Projects folder.

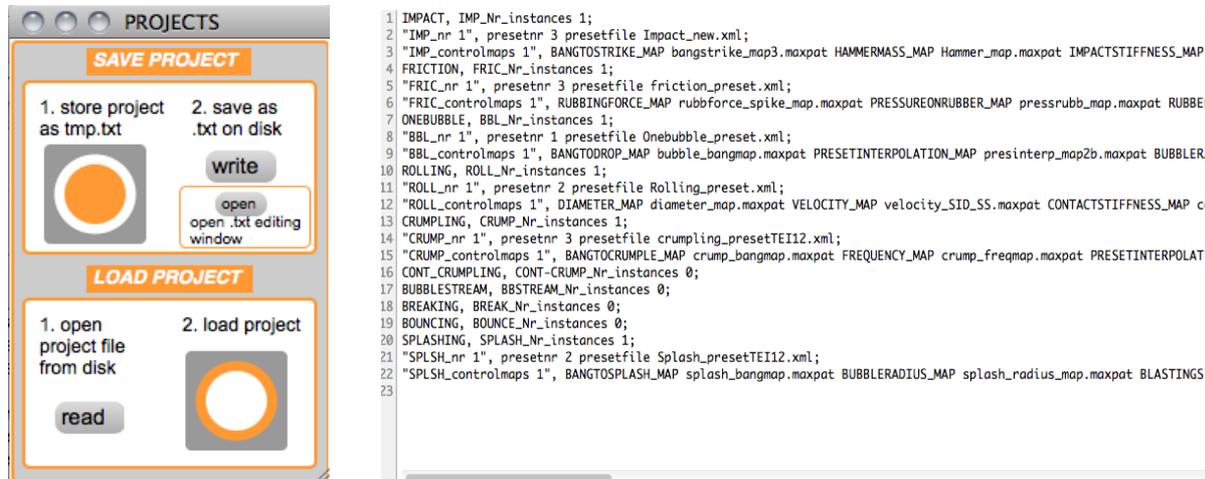


Fig. 11: Snapshots of the SDT environment are saved on disk as .txt file.

References

[1] Delle Monache, S., Polotti, P., and Rocchesso, D. (2010). A Toolkit for Explorations in Sonic Interaction Design. In *Proceedings of Audiomostly 2010, a Conference on Interaction with Sound*. Pitea, Sweden. September 15 – 17, 2010. (available at <http://rice.iuav.it/170/>)

[2] Delle Monache, S., Devallez, D., Drioli, C., Fontana, F., Papetti, S., Polotti, P., & Rocchesso, D. (2009). Sound design toolkit – Users' guide. In *Closing the Loop of Sound Evaluation and Design (Deliverable 2.3, pp. 1-62)*. Verona, Italy: University of Verona (available at http://closed.ircam.fr/uploads/media/CLOSED_D2.3.pdf)

¹ For instance, the “Gamelunch: a sonic dining table” (<http://vimeo.com/874774>) and the “Sonic paper vessel” (<http://vimeo.com/36679365>) which made extensive use of several sound models.