Patch Pd para Nih-Nik de Chico Mello[1]

Gabriel Montechiari e Silva[2]

Federal University of Paraná, Brazil

This *Pure Data* patch was commissioned by the Ensemble Nih-Nik for the premier of the homonymous 1993 piece by Brazilian composer Chico Mello at the SiMN 2014 + *matrix14 on tour* festival (Curitiba, Brazil). The piece was written for trombone, percussion, and electronics that consists of: a harmonizer for the trombone with four different settings (including off) switched by the player; and a sampler to be played by the percussionist.

The score required musicians to move on stage, so the choices of hardware were two buttons attached to the trombone and connected to the computer via an Arduino board, and a MIDI controller mounted on a wooden support attached to the percussionist's upper body. The harmonizer and sampler sounds were played through two loudspeakers; one positioned behind the musicians the other behind the audience.

Two *Pure Data* external objects were used in this patch. The first one was the *Arduino* object from *Pduino library*, to allow the interface between the Arduino and the program. The other one was *pitchshifter~*, since there were technical issues regarding the *freqshift~* object. Objects from the *list-abs* library were used as well, but those are already part of the Pd Extended 0.43.4 bundle. The patch was made using the Mac OS X version of the software.

On the harmonizer bit of the program, as the GUI's Set button is pressed, the Arduino object receives messages to configure the board's pins 2 and 7 as inputs. Once this is done the buttons can interface with the software. The bangs sent by them pass through a piece of patch that allows only one bang each 500 milliseconds to avoid any accidental jumps in the program, as one button increases a count upward and the other decreases it. This count represents the index in a list of harmonizer presets. Each harmonizer preset is another list containing three transpose values (the unit represents half a tone,

[1] Submitted on: 11/01/2014. Accepted on: 12/01/2014.

[2] Email: g.monte.silva@gmail.com

zero stands for voice off). The active preset list is sent to an object that counts how many zeros there are and later multiplies the signals proportionally, so the resulting volume is always in unity gain. Then three *pitchshifter~* objects receive one list value each (excluding zero, which cuts the signal to that specific *pitchshifter~*). After that, the signals are summed and directed to the digital to analog converter.

On the sampler bit, the MIDI notes are sent to a *poly* object, to enable polyphony. Each note is then routed and sent to an abstraction called *tocaNota*. In this abstraction there is a *readsf~* object that opens and plays the sound files as the notes are pressed, as well as a conditional statement that determines which loudspeaker will the sample come out from: front, rear, or both (according to the score). There is also an envelope to avoid clicks when the note is lifted and a piece of patch to normalize the sound (unity gain if sound comes out from only one speaker and half the gain if it comes out from both).

After the premier, one issue stood out. The trombone player had trouble keeping track of the count. In future editions I intend to add two more buttons so the count can be dismissed and the harmonizer presets switched directly. Also, the piece requires a modulation wheel to control the sampler's pitch. This feature will be available in a revised version.