Computer Music - Languages and Systems

Group 5: PLAS

Assignment 5: Granular synthesis for foley sounds

Homework 1 Report

Students

Alberto Doimo
Paolo Ostan
Stefano Donà
Lorenzo Curcio

POLITECNICO
MILANO 1863

Description of the problem

This homework assignment aims at the creation of different Foley sounds with *Supercollider* audio development environment. In the specific, the technique used for the generation of sounds is called *granular synthesis*.

Granular synthesis algorithms usually start from samples/sounds and operate a sampling onto them in order to generate short audio clips called *grains* that are then layered or combined in different ways to form new sounds.

In addition, the synthesizer developed provides a graphical user interface where the sounds could be played (alone or together) and where the user may decide how to mix them, and how to set their behaviours by manipulating some different parameters regarding sound generation, post-processing and perception.

Approach to the problem

The typology of problem needed some preliminary decisions before the implementation:

- Given the great possibility of manipulation to the generated sound offered by the environment used, in contrast with the specific characterization of sounds used in *Foley* technique, two strategies were considered:
  - One in which the user may widely manipulate the parameters related to the processing and generation of the sounds, leading to possibility of unpredictable and maybe unwanted results during development.

  - One in which the system limits the capability of manipulation of the user, masking to the user the manipulations done to the sound in order to obtain a more controllable and predictable end result.

  After some discussions and tests the second strategy was adopted, in order to better maintain the whole project direction.

- The necessity to allow the user to interact with the different sounds in some way lead to the creation of an interface that controls the position in space of the various sound elements. This led to the mapping of some exposed parameters in order to allow the user to control them through the *GUI* [1].

- Finally, the group decided that the end result would benefit from having a common theme that was shared between the different sounds we would synthesize, and same can be said about the possibility to interact with the different sounds in some way.

Sound synthesis

The sound synthesis was performed using the same few seconds of music for all of the four sounds. The decision has been to create a forest-themed soundscape where four sounds could be played: fire, the howl of a wolf, footsteps and the cry of crickets and frogs (these last two play together).

To synthesize the sounds, primarily a very helpful *UGen* contained in *Supercollider* called *Grainbuf* was, it can operate granular synthesis of sounds stored in a buffer.
For each of the four sounds, the group selected some parameters to be connected to different knobs. Firstly it was considered the possibility of adding fully-customizable synthesis parameters and ADSR envelope to each of the sounds, however the result ended up sounding very different, and often very artificial from a Foley-oriented perspective. For this reason the manipulation

[1] Graphical User Interface

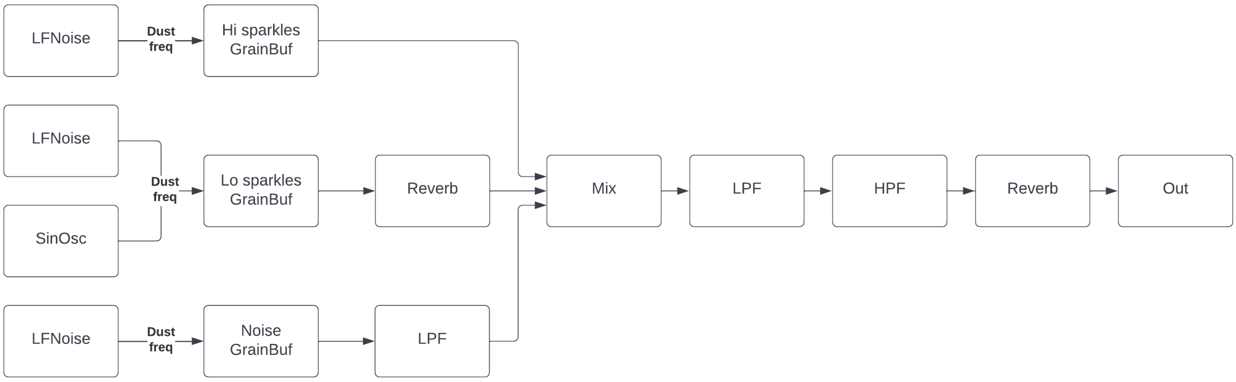

Figure 1: Fire Signal Flow

possibilities were limited.

Fire

For the fire sound the basic idea was to create three different signal sources. The first is a background noise obtained by using very small grains triggered by casual impulses at a high frequency, using the UGen *Dust*. To give a better realism to this noise both modulations in frequency and amplitude were applied, thanks to the use of *LFNoise*. This noise was also low-pass filtered with a modulated cut-off frequency centered at 4kHz. A second signal aims at creating a crackle sound, keeping the grains duration short (but greater than the noise ones) and using a low frequency *Dust* UGen as trigger. The last signal is quite similar to the previous one, but has a lower amplitude, higher trigger frequency and it is sent into a reverb that lengthens the sparkles tail. The three signals are finally mixed together and filtered using *HPF* and *LPF*. (fig. 1)
A "Wood" knob controls how often (on average) a crackle occurs while the more general sound of fire burning keeps playing in the background. This has been made possible by connecting the wood parameter to the amplitude of the noise signal and the frequency of *Dust*s UGens.

Wolf

The wolf synthesis required a different approach due to the necessity of a sustained sound. The idea was to obtain a tonal signal characteristic of wolves and to do that we chose a grain length of around 0.3 seconds triggered by a train of impulses with a modulated frequency around 350Hz. A sinusoidal envelope also contributes to the frequency modulation. The grain rate has been kept low (0.2), in order to lower the harmonic content of the original buffer, the same rate is also regulated by an ADSR envelope that brings up the tone in the attack-decay phase, stabilizes it in the sustained part and finally releases it. The whole signal is finally modulated in amplitude using a *LFNoise* and the ADSR envelope, low-pass filtered and sent to the output bus. (fig. 2)

For the howls, we made a "Length" knob that controls how long the wolf will play each time acting on sustain of ADSR envelope, a "GrainPos" knob that sets where to extract the grain from inside the sample, and finally an "Occurrence" knob that just controls how often the sound

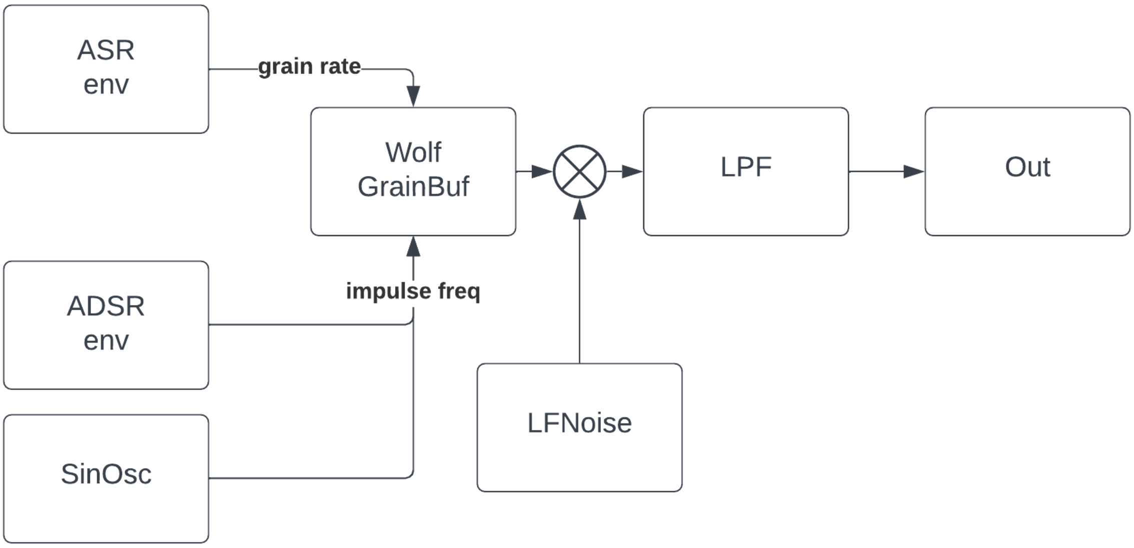

Figure 2: Wolf Signal Flow

is heard simply increasing the frequency of a trigger that acts on the envelope.

Footsteps

For the footsteps, the processing was much simpler, a specific grain obtained from the sample was multiplied by a signal made using *BrownNoise*, while its trigger function was chosen to be a regular train of pulses to which a random variation was added. We made a "Dirt/Snow" knob that controls the frequency at which a high pass filter is applied, making it sound like the steps are taken on a snowy ground when heavily filtered and on dirt when left untouched. A second knob changes the frequency of the steps, similarly to the previous cases. (fig. 3)

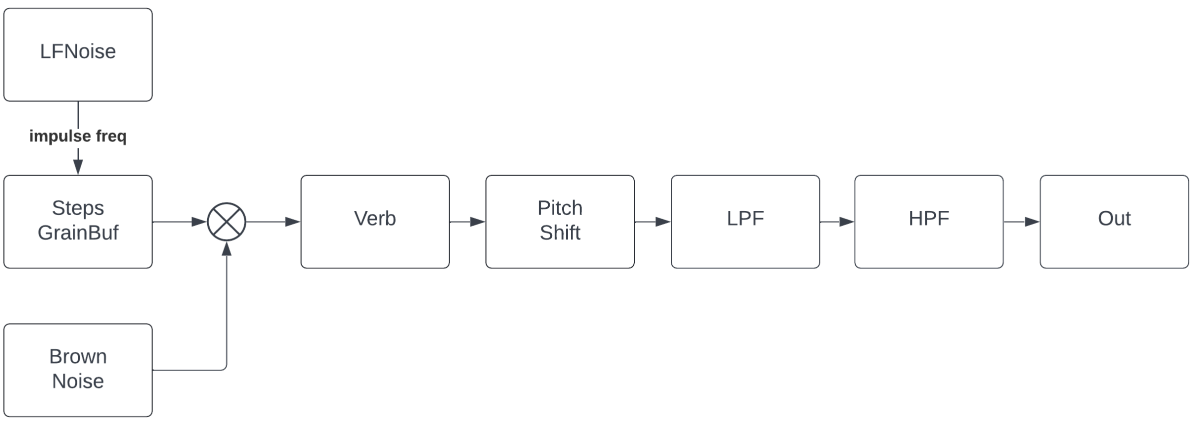

Figure 3: Steps signal Flow

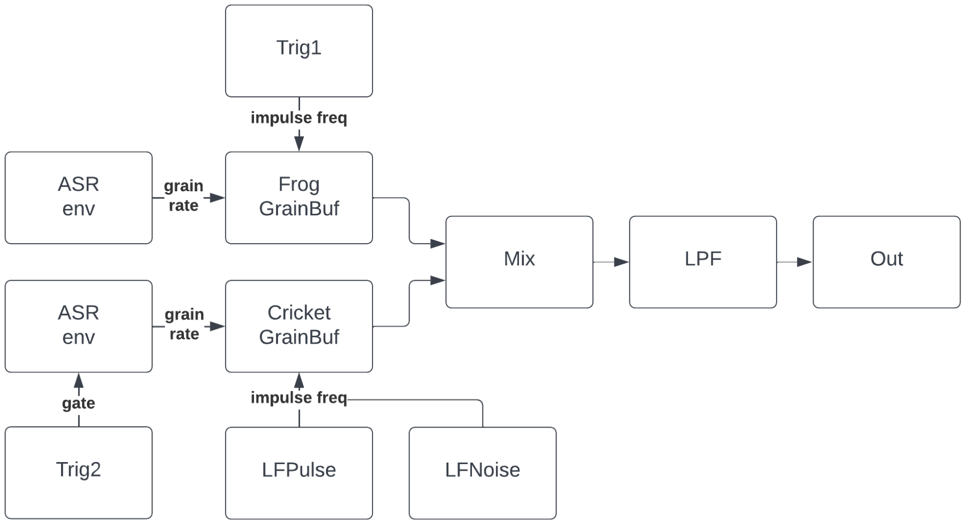

Figure 4: Frogs/Crickets signal Flow

Crickets-Frogs

The crickets-frogs synth was conceived similarly to the fire one but with some differences. The grain duration is kept around 0.01 sec, in order to obtain a fragmented sound. The synth is basically composed by two main signals. The first represents the frogs, and its grains are played thanks to a 30Hz train of impulses triggered by a *Trig1* UGen with a duration of 0.3 sec. The play rate of the grains is 5, in order to obtain high pitch and it's also modulated by an ASR envelope.
The second synth represents the crickets and the construction of this signal is very similar to the frog one, except for the rate, taken to 15 and modulated by a triggered ASR envelope, and the impulses' train frequency, taken to 60Hz and modulated by an *LFPulse* at variable frequency. Both signal are finally mixed and low-pass filtered at 1500Hz.(fig. 4)

Finally for the crickets and frogs, we made a "Randomness" knob that controls how irregular the cries are, acting on frog's trigger frequency, and two "Excited" knobs that control how often and loud they are played, just acting on the amplitude of each signal and frequency if triggers.

The ranges of the values are listed in the next table.

Spatial Position

In order to allow the user to set a personalized soundscape image, the group decided to implement a two-parameters controller that gives the possibility to move the sounds generated inside the stereo field. This feature is built in such a way that the user may control both the direction from which the sound is heard, while being able to modify the distance as well. This is allowed through the use of a Synth called *Positioner* that processes the mono output provided by each *Sound Generator* and outputs two stereo busses, one of these is used to send the dry signal to the master stereo output. The second one is used to send the stereo signal to a reverb effect, in order to add cohesiveness to the soundscape and to have the user perceive a more immersive

Knobs	Lowest Value	Highest Value
Wood	0.5	10
Length	2.5	5
GrainPos	0	1
Occurence	0.1	0.5
Randomness	0	1
Dirt/Snow	200	2000
Speed	1	2
Excited 1	0.5	1.7
Excited 2	0.5	1.7

Table 1: Values of the parameters

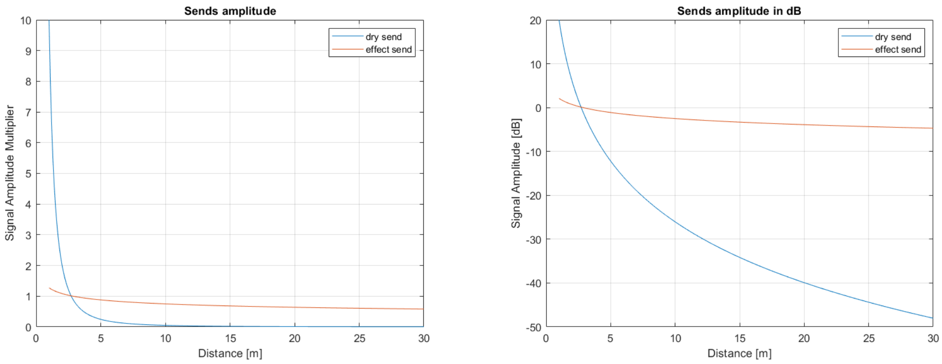

Figure 5: function for the main volume

environment.
The two outputs are characterized by two volume curves (fig. 5) that simulate the distance dependency of a sound source. The volume of the dry signal follows the logarithmic law of attenuation of sound, starting from a virtual value of 20dB at distance 1 m. The maximum distance we set is 30 meters from the listener, and the sound pressure level undergoes a 6dB attenuation each time the distance gets doubled. Some research and testing on the parameters led to a quasi-constant relation between distance and volume for the wet signal in order to better mimic the response of an acoustic environment.

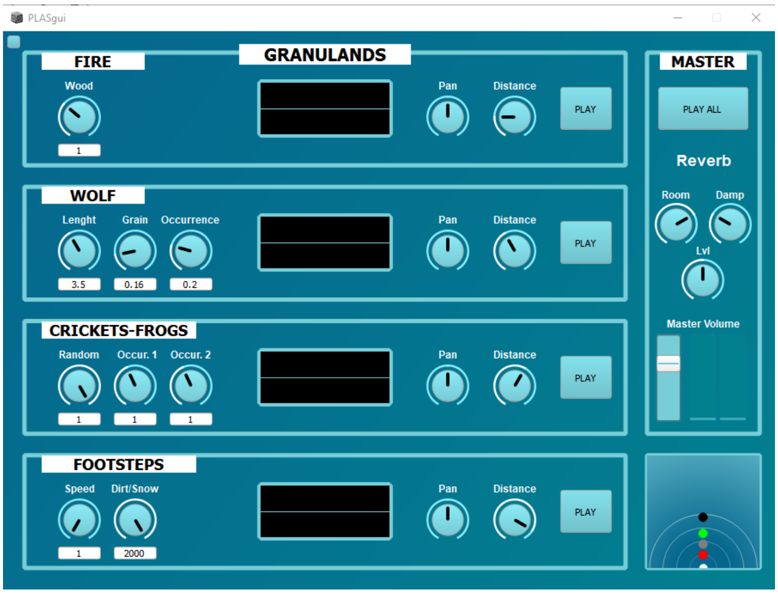

Figure 6: Main window

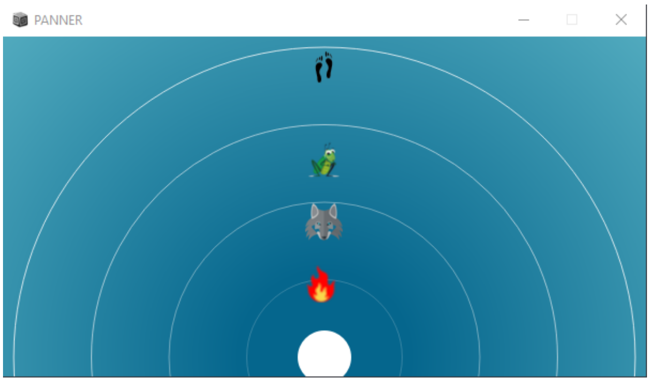

Figure 7: Panner window

GUI

The graphical user interface is composed of two windows: the main window to mix the different sounds and a window to visualize and control the spatial position of the elements in the stereo spectrum. In the former there are four sections (one for each sound), each one with the knobs that control the different parameters discussed previously. Moreover for each sound there are two knobs (pan, distance) that control the position of the sounds and a play/stop button as well. In the center of each, the oscilloscopes allow to monitor the outputs' waveform. On the right the master controls are present: three knobs used to manipulate the reverb applied to the master channel, a Volume slider that allows to set the general volume and a level indicator to visualize the output volume. Furthermore, on the lower right corner a minimized version of the second window is present, containing the spatial positions of the elements, represented as coloured dots. Clicking with the mouse on the view the second window opens. Finally a button in the top left corner can switch the backgrounds on and off, useful to give a visual feedback type of *Foley* sounds generated. The second window contains the different icons for the different sounds, and can be used to drag them in different positions to trigger the spatial audio parameters and to visualize the spatial position of the objects when changing the parameters through the use of the knobs.

Results

The subject matter allowed the experimentation of sound generation through the use of *SuperCollider*. Specifically, the technique of granular synthesis was used for a different purpose than the conventional ones, like creative sound design or other music related purposes.

The results obtained allowed to underline the combination between creative work and programming, needed for sound generation and manipulation. The result is a set of four sounds that aim at emulating what one would hear at night in a forest, with minimal and specific controls on sound generation and some more general ones on spatial sound perception. The Graphical user interface completes the result and allows the user to interface with sound manipulation in a more intuitive way, matching the modification on the sources and on the overall soundscape with a visual feedback.

Future Development

Below are listed the possibilities for future development of the project.

- Adding a greater number of sound sources, and consequently developing a more complex *Graphical user interface* in order to show and control them
- Expanding the spatial audio processing implementing a complete ambisonic processing environment in order to develop an immersive experience with extended control capabilities for the user.