Computer Music - Languages and Systems
Homework 1
Fourier's phone: Additive Synthesis

Group 6 - GELVS:

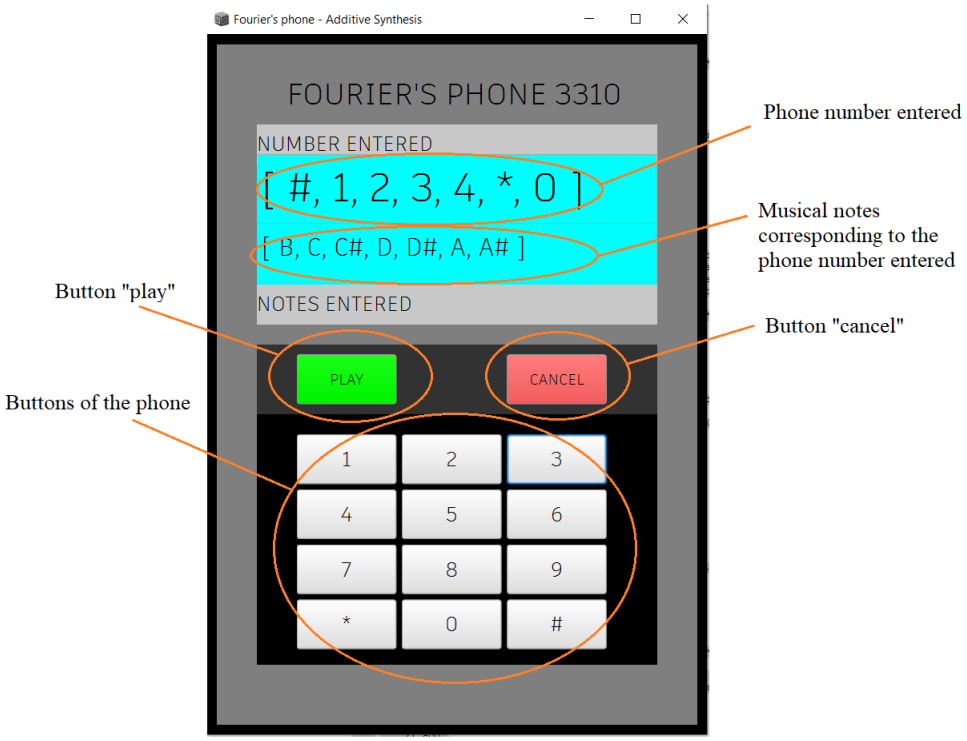
Del Moro Samuele, Di Clerico Letizia, Negroni Viola, Perego Gabriele, Roncuzzi Enrico.

1 Goal

Create an instrument in SuperCollider based on additive synthesis and a simple interface to control it, using either an external device or implementing a sequencing technique with SuperCollider to play it. Regarding the control part, implement a user interface where it is possible to control the synthesis parameters.

2 Concept

If Fourier had lived in the 90s, how would he have reacted in front of a mobile phone full of ringtones? We created an instrument (with the appearance of an old mobile phone) able to create ringtones through additive synthesis, generating a series of sound waves that can be played individually or simultaneously, and which can be modulated by changing some of their parameters.
A visual feedback of the resulting sound is given by both a frequency analyzer and a stethoscope.



3 Method

3.1 Overview of the GUI

3.1.1 The mobile phone

The mobile phone interface, as can be seen in the figure above, is designed to be simple and intuitive. In the top part of the window a screen can be immediately recognized: here is where each pressed number will appear, along with the corresponding musical note.
At the bottom instead, a basic keypad is placed, while right on the top of it the "play" and the "cancel" buttons can be found.
Note that these two keys will be grey and blank at first, the colors and their names will show up as soon as the user presses a key.
Every time a button is pressed in the keypad the phone will also generate a sound.
It is possible to compose a "phone number" sequence from 1 to maximum 7 numbers (with the corresponding musical notes), so that it is possible to generate, for example, chords, scales, etc...
The pattern used to associate the phone keyboard symbols and musical notes is very simple and intuitive.

The structure is the following:

PHONE SYMBOL	1	2	3	4	5	6	7	8	9	*	0	#
NOTES	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
fundamentalNote = 523.20 ---> C5												

where *fundamentalNote* is the reference variable for the pattern, which can be arbitrarily modified.

3.1.2 The synthesizer

Once the play button is played on the telephone, the additive synthesis window will appear, side by side with a frequency analyzer and a stethoscope.
The window is basically divided in four parts, from the left to the right the user will find: a set of 6 knobs for every chosen tone, through which it is possible to modify the default parameters of the sound wave:

- the detuned frequency factor: it increase (or decreased) the frequency of the tone by a maximum of 50 Hz.
- the amplitude: starting from a default value of 0.5, can be brought down to 0 or up to 1.
- the speed: 10 different speeds are allowed, the default value is 1.
- the amplitude modulation: 100 values allowed, starting from 0, up to 10, the default value is 1.
- the range: in which the frequency can vary between the main frequency and its multiples, 5 values allowed.
- the octave: it is possible to go up (or down) two octaves.

Then, next to the knobs, we can find the last three sections.
The first one contains the labels of the originally chosen note (that also serve as play/stop buttons for the sound), the second one states the selected waveform (default is the sine, but it is possible to change it by clicking on the button when the sound is not playing), and the third indicates the current fundamental frequency of each wave (it will change only accordingly to the turns of "detune" and "octave" knobs).

3.2 Code implementation

3.2.1 General structure

As regards the organization of the code, in order to better delve into it later, we shall here state that it can be seen as roughly divided in three main conceptual sections: a declarative part at first, the

phone configuration in the middle, and basically the operative core at last.

1. The global variables, the SynthDef and the main function are here defined:

   - Variables: among them, note that we here set the fundamental frequency at 523.20 Hz, the frequency of a C5.
     This will be the starting point to build the other notes.
   - SynthDef: four SynthDefs are defined, the first is dedicated to the sound emitted by the pressed phone keys, the other three are meant to set up the 3 different types of waves (details in section 3.2.2).
   - Functions: the functions "e", "f", "g", "h" are basically dedicated to the generation of the notes, setting the frequencies, and to associate them both with the right phone keys and displayed labels.
     The functions "funPrintText" and "funPrintKnobText" print respectively the pressed numbers with the corresponding notes on the phone screen, and the labels of the customizable parameters in the additive synthesis window (details in section 3.2.3).

2. As it is also suggested by the comments in the code, this part is dedicated to the set up of the phone layout, from the backgrounds to the buttons, to the rendering of the typed string.
   Note that the set up of the second interface is included in the third section (see section 3.2.4 and 3.2.5).

3. Here the setup of the additive synthesis window can be found, as well as the implementation of all the functionalities made to make it work eventually.
   This whole execution is basically subject to the pressing of the "play" button on the phone (see section 3.2.6).

3.2.2 SynthDefs

A SynthDef is basically a synthesizer, its aim is to produce sound waves.
As previously mentioned, four SynthDefs are employed within the project: the first one, "ring", is more related to the overall appearance of the telephone, since it generates the sounds accompanying the pressing of a key, while the other three ("addSin", "addSaw", "addTri") will generate the three different kind of waves that it is possible to implement while setting the parameters for the sound synthesis.

Here's how they are built:

- "ring": the key tones are produced by a pulse oscillator (LFPulse.ar) which is controlled by an envelope (Env.new), whose values for the levels and the controlling times we arbitrary set.
  Note that doneAction equals 2 in order to automatically deallocate the Synth.
  The envelope prototype is then generated through the prototype (EnvGen.kr) at control rate, since it is not actually creating sound, but just controlling the pulse oscillator.

- "addSin/addSaw/addTri": their implementation is the same, the only thing that changes is the waveform.
  We specify the default values of the arguments.
  The first five refer to the ones that will be also possible to modify in the additive synthesis phase: amplitude (totalAmp), fundamental frequency (fund), amplitude modulation (ampHz), speed (vel), range in which the fundamental can possibly vary (maxPartial).
  Both the output channels (left and right) generate the sound wave (SinOsc/LFSaw/LFTri), whose amplitude and frequency are designed to be variable, so that they can be changed by the user through the knobs during the synthesis.
  The amplitude modulation is described by a sinusoidal variation, governed by ampHZ, around the given amplitude value.
  SinOsc is at control rate, since it is not meant to play any sound, same as LFNoise0 for the frequency variation.
  This last scenario will take place if the user sets the "range" (maxPartial) to a value bigger than

zero: the actual frequency of the signal will start changing randomly, with speed "vel", in a range between the fundamental and the same fundamental multiplied by the chosen value (only taking as feasible values integer multiples of the fundamental in that range).

3.2.3 Functions

Six main functions were defined in order to allow the typing of numbers and notes on the telephone:

- Function e: converts the array of numbers entered to a string array of musical notes.
- Function f: takes the numerical value corresponding to the button pressed and uses it to calculate the power of two corresponding to the frequency of the corresponding note.
- Function g: maps the pressed button with the corresponding musical note.
- Function h: maps the pressed button with the corresponding symbol.
- Function funPrintText: prints on the screen the numbers pressed and the corresponding note.
- Function funPrintKnobText: prints the name of the variable parameters of the waves above the respective column of knobs.

There is also another function (function egg) which, based on an input array, creates and shows a particular window (see section 7).

3.2.4 Interface configuration

The interface configuration starts with the initialization of the parameters of the main window.
Specifically, the name, the edges and the main settings have been set.
If the window is closed, all nodes registered on the server are released.
Then, in the same way, the parameters of the phone screen were set and static text was added.
The main properties that have been set are the size of the windows, the font used and the colors.

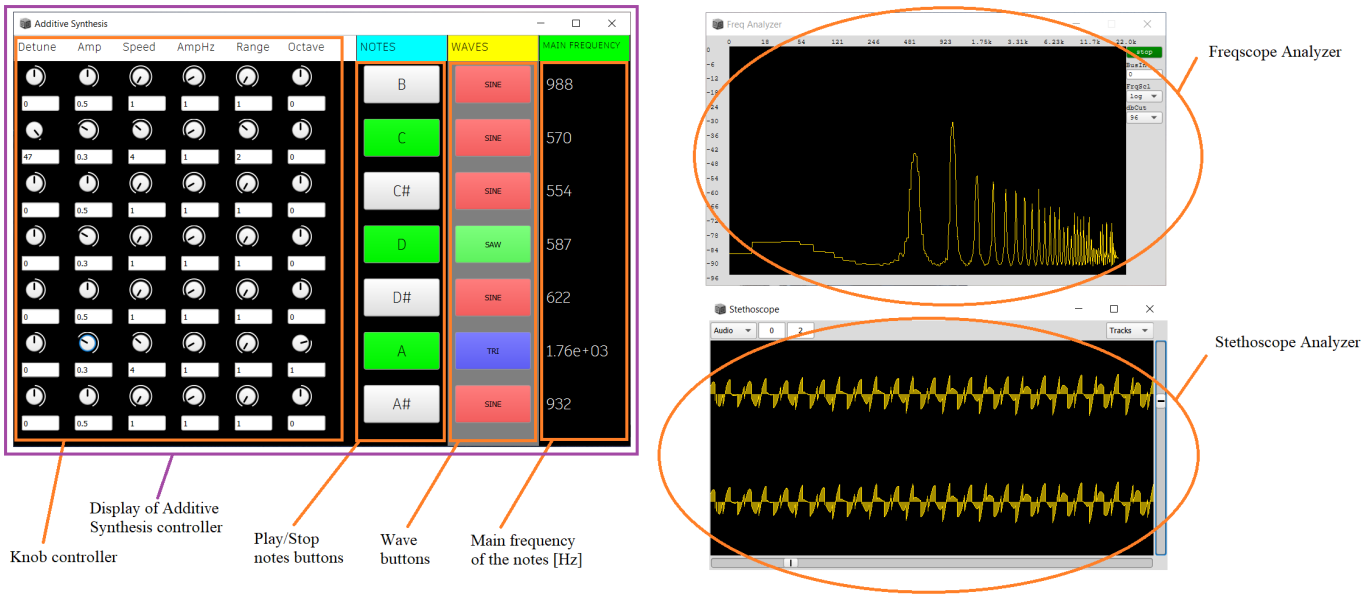
3.2.5 Telephone keypad

The telephone keypad is made up of a set of 12 buttons in which the numbers from 0 to 9 can be found, the asterisk $*$ and the hash $\sharp$ are set.
Above the actual keyboard there are two other buttons, initially disabled: the "playButton" and the "cancelButton", whose color will turn respectively green and red after the insertion of the sequence of numbers.
After proceeding with the setting of the parameters (shape, color, position and the content at the inside) for every components, every element has been programmed individually.

We can divide the elements into three parts, in order to explain the individual behavior of each one:

- "cancelButton": check the numbers entered on the phone by means of a counter. If the number of elements is greater than 0, it deletes the last one element, if necessary until they are all deleted. If the number of elements becomes 0, both the play and cancel button are disabled and turn grey again.
- "keypadButton": when a keyboard button is pressed, a synth ("ring") object is generated with a frequency relative to the relationship between numbers and notes: a short sound will accompany the pressing. At the same time, the number pressed will appear on the display (if the numbers entered are less than 7). As mentioned above, if the number pressed is the first one, the playButton and cancelButton are activated.
- "playButton": when the "playButton" is pressed, the additive synthesis window opens. A detailed description of the aforementioned window can be found in the following paragraph.

3.2.6 Additive synthesis window



If at least one number is typed, the button "PLAY" turns green and, once pressed, a series of actions begin:

- The window that contains the telephone becomes invisible and two scopes are set up, in order to give visual feedback of what is happening to the waves.
- Initialization of all the displays that contain all the knobs and the buttons that will be needed.
- Dynamic initialization of the matrix of knobs, needed to allow the user to set the 6 parameters as he/she desires.
  The matrix dimensions are N x 6, where N is equal to the number of symbols (notes) previously typed.
  An iteration is performed on the knobs, setting their value one column at time.
- Visualization of the frequency range of the waves on the same window.
- The states of start and stop for the buttons are created.
- The buttons to control the type of wave to be played are set up. Every time a button is pressed on the column "wave", the corresponding SynthDef is set. Note that the user can't change the shape of the wave as long as the wave is active (hence, until the button is turned off).
- The last column "main frequency" indicates the reference frequency of the chosen note. It is possible to change its value only if the reference octave of the note is changed by the "octave" knob, or the note is changed slightly in frequency by the "detune" knob.

3.2.7 Plots

In order to better appreciate the resulting sound in visual terms, two simple tools have been implemented within the additive synthesis phase:

- A frequency analyzer window, showing the spectra of the selected tones and how they possibly overlap with each other
- A stethoscope, showing the temporal progression of the generated sound, hence allowing us to visual feedback on the chosen shapes of the waves.

Since we considered these plots to be already effective for their purpose, both of them are left with their default settings.

4 Results

Accordingly to the implementation discussed above, the user-experience of this project has been designed in order to guide the user through two different subsequent steps:

- The composition of the 'telephone number', which results in the setting of the number of sound waves to play with, along with the choice of their starting frequencies.
  We here recollect that the maximum length of the string is 7, hence it is possible to generate a maximum of 7 waves at the same time, whilst the predefined notes start from a C5 for the key 1, moving up by a semitone for each keypad number.
- The additive synthesis stage, where it is possible to play with the selected waves by changing:
  - The shape of the sound wave (sine, saw-tooth, triangular)
  - The frequency (the starting one being determined, as stated above, by the pressed number)
  - The amplitude of the single waves
  - The amplitude modulation of the single waves
  - The range in which the main frequency can vary
  - The speed at which the frequencies of individual waves change

  These modifications can be performed independently for each component, and it is also possible to start and stop them independently as well.
  The resulting ringtone can be appreciated, other than of course listening to it, by visualizing it through a frequency analyzer window and a stethoscope.

5 Conclusions

This project proposal has been conceived with the aim of interpreting the given task with a humorous cut, in order to make the user-experience engaging and enjoyable.
Nonetheless, beside the playful fashion, each element has been implemented always keeping an eye on the ultimate goal: to develop a simple yet effective tool to perform additive synthesis, making it as interactive and customizable as possible: this is achieved by designing a rather basic GUI, which makes easy and immediate to explore its functionalities, and preferring a compact yet rich view also for the additive synthesis window.

6 Further Improvements

Some potential and further improvements could be:

- Improve the graphical interface.
- Give the possibility to insert more than 7 elements, according to the user's needs.
- Add other parameters to modify the behavior in the generated sound waves.
- Add other types of waves (e.g. LFPulse, PinkNoise etc...), giving the possibility to change run-time the type of wave during the execution.

7 Easter Eggs

To make the user's experience more enjoyable, some Easter Eggs can be found when the telephone sequence is entered: depending on a specific entered number sequence, through the function "egg", some simple windows, with different colours, will appear in the center of the screen, which will notify the discovery and the type of Easter Egg found.