Computer Music - Languages and Systems

Group 5: PLAS

Homework 3 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 a *Computer Music System*, starting from its design, to its implementation. In particular, the system would be made of an interaction unit, a computer music one, and a Graphical User Interface.

To do this, some preliminary decisions had to be made. Firstly, the group had to figure out a context where a particular activity could be monitored by making use of the available tools, and where the acquired data could be mapped in a meaningful way to control some predetermined musical parameters. Secondly, the connection of this mapping to the music had to be finalized. Finally, a way to reflect the different perceived events visually in the GUI was needed.

Approach to the problem

Given the task, the group decided on an activity that could offer many different layers for measuring or monitoring. This was done in order to guarantee many possible ways of generating data that could suit a computer music system. For this reason, the final decision was to generate music using the movements of a basketball player.

The communication of the different components is built as follows: the shake sensor communicates through an Arduino control device with Processing using OSC messages, while the accelerometer and the gyroscope of the smartphone (collected using *MultiSense Osc*) directly send OSC messages to Processing. Finally, after being mapped through Processing, the data is sent to Supercollider, which uses it to modify the loop.
The way the system works can be seen in the scheme in figure 2, while in figure 1 all the different devices are represented.

Music

The music, generated using Supercollider, is intended as a background for a training session, and it's based on a continous loop made of drums patterns with a 4/4 time signature, a pad which establishes the harmony through a random selection of many pre-made patterns, and more pre-made melodic patterns for both a pluck synth and an electronic bass synth. All of the patterns are written on the melodic minor scale, whose root note can be selected through the GUI as explained in the following sections. To extract data from these movements, a shake sensor and a smartphone were used.

Figure 1: Devices scheme

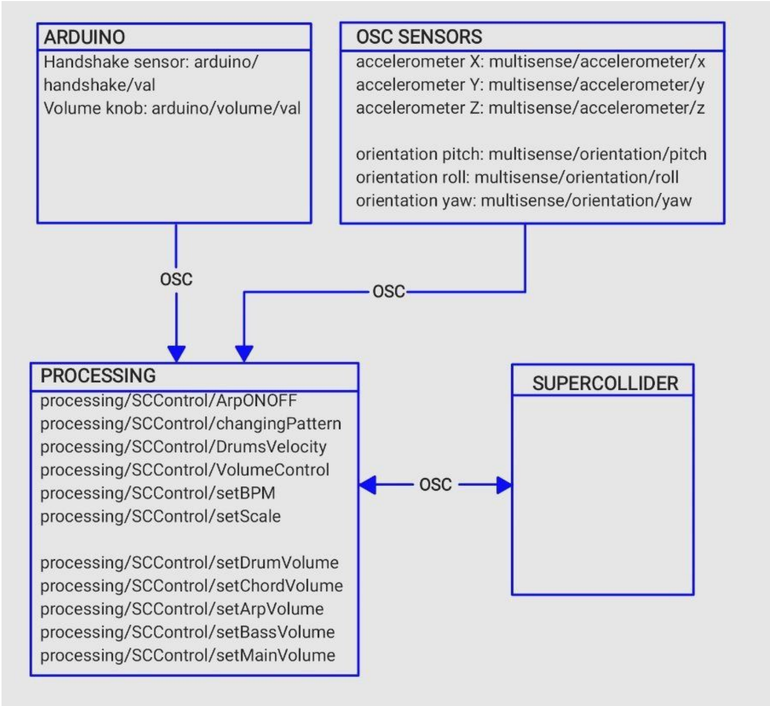

Figure 2: Communication diagram

Experimentation

In order to select which movements and sensors to use, the group had to experiment with different configurations. The final choice of using a shake sensor, an accelerometer and a gyroscope is the product of several testing sessions that underlined which type of data was more suitable, and above all how to process it in a meaningful way. This choice was also motivated by encumbrance considerations, since a basketball player can easily keep his smartphone inside his pocket and install the shake sensor on his main hand.

For example, the heart-rate sensor ended up being discarded due to the monitoring being very unstable while moving, even at very low speed. On the other hand, the shake sensor proved itself much more reliable to calculate rates, due to it having a discrete and dual behaviour instead of a continous one.

For what concerns data processing, each of the three sensor were mapped as selectors for different states of the drums, the arpeggio and the bass patterns, as described in the following sections.

Acceleration

This sensor provides acceleration values in all of the three cartesian coordinates, that are then post-processed to compute the magnitude. This value was then mapped using different ranges, as seen in table 1, to be a selector for different types of arpeggios, so that its speed and complexity increases according to how fast the player moves in space.

Orientation

In order to better follow the player's movements while playing, monitoring the orientation as well as the acceleration turned out to be a good solution. The outputs of *MultiSense Osc*'s gyroscope are roll, pitch and yaw. Considering the position of the phone in the player's pocket, it was decided to map the yaw parameter. The gyroscope ranges from 0° to +360°, therefore this interval was equally divided in five sectors that are associated with five different bass melodic patterns as seen again in table 1.

Shake

As previously mentioned, the shake sensor turned out to be one of the easiest sensors to map. The amount of shakes occurring in 1.5 seconds is registered and mapped to control which one of the three drums patterns to use. The patterns are ordered from the calmest to the most chaotic. This way, by placing the sensor on the player's thumb, the player itself controls how strong the rhythmic component feels, based on the basketball dribbling rate.

Sensors	State 1	State 2	State 3	State 4	State 5
Accelerometer $\left[\frac{m}{s^2}\right]$ (*Arpeggio*)	≤ 12	$12 - 14$	$14 - 16$	≥ 16	
Orientation [°] (Bass)	$0 - 72$	$72 - 144$	$144 - 216$	$216 - 288$	$288 - 360$
Shake Sensor [shakes] (Drums)	0	$1 - 2$	≥ 3		

Table 1: Values of the parameters

Arduino

This project was developed using a *MKR WiFi 1010* Arduino board. Thanks to the WiFi module, the player can move around freely.
The board is powered by a micro USB port using a small 5V Power Bank that powers the shake sensor, the potentiometer and an RGB led as well. The system was created and developed on a breadboard (as seen in Figure 5), but it has been moved to a Stripboard in the final version to have a more robust system. Given the working conditions of the system, the group used an enclosure case to host it, so that it can be easily installed on the player's arm.
Pictures of the system can be seen in figure 3 and 4.
When connected to the Power Bank, the led flashes purple until it connects to the network. Then, when the connection is engaged with the PC, the led flashes six times (RED-GREEN-

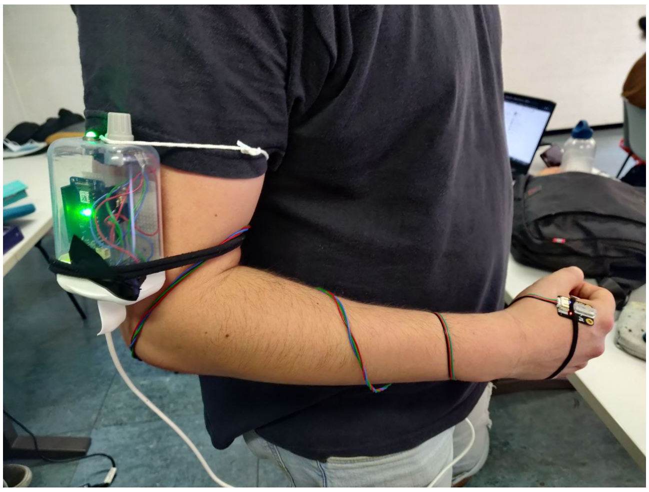

Figure 3: Mounted system

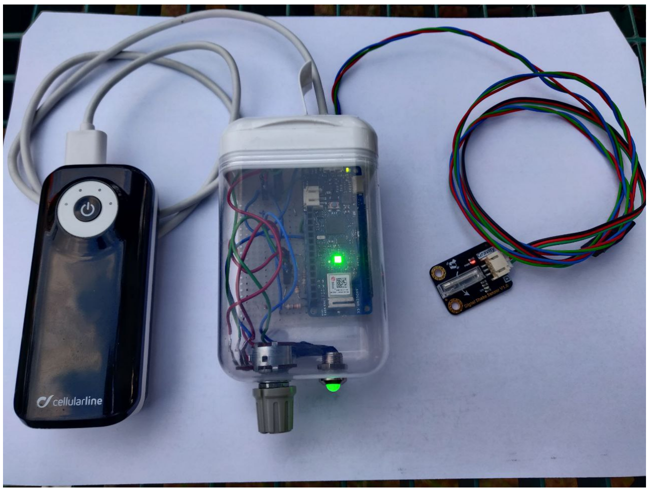

Figure 4: Unmounted system

BLUE-GREEN-GREEN-GREEN). When the system is working the led stays GREEN and its intensity varies with the master volume.

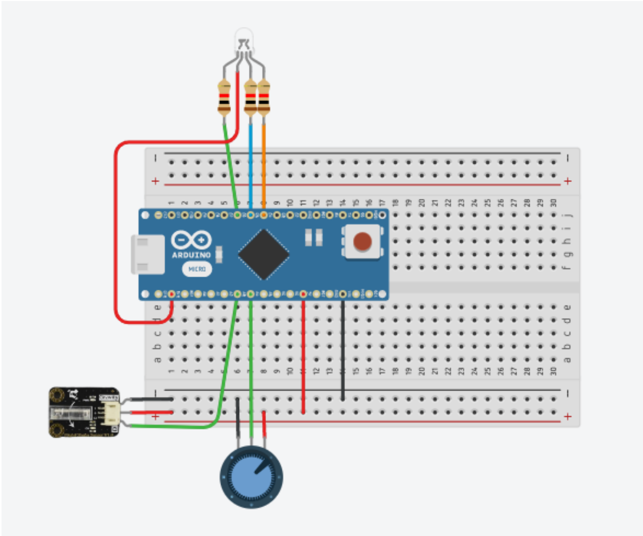

Figure 5: Arduino configuration

GUI

In order to visualize the interaction between sensors and sounds the group developed a GUI using Processing. This allows a graphical visualization of the various parameters' evolution, using basketball related themes. The group developed a 3D interactive ball that changes its shape and behaviour according to OSC messages received by Processing and to the audio signal received by Supercollider, so that the user can have a visual response about the music that is played. To achieve this, the output from Supercollider was sent using four channels in an internal routing: two of them are sent to Processing in order to control the ball's movements, while the remaining two are directly sent to the audio output. In particular, the shake rate is used to control the rotational speed of the ball, the radius of the ball increases with the amplitude of the input signal and the background lines move when triggered by a beat detector. The final look of the main window can be seen in figure 6.

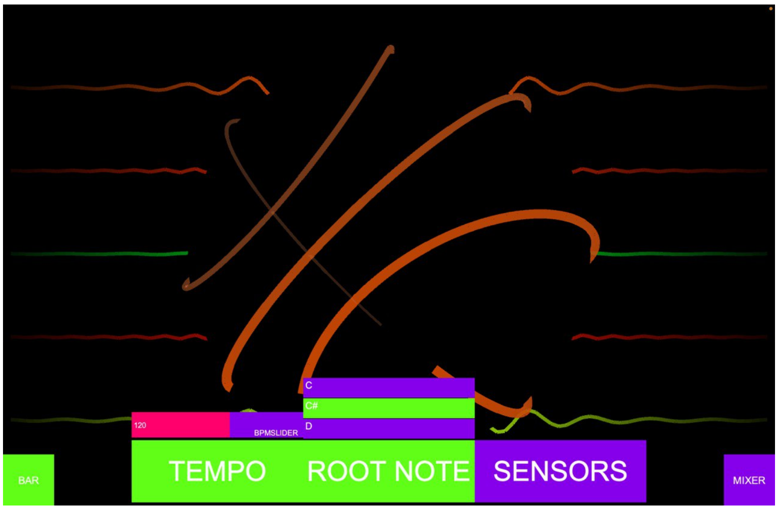

Figure 6: GUI Main Window

In the lower part of the main window there is a tool bar comprising a Tempo button, in which the BPM value can be set by a slider, a Root Note selector and a Sensors button. The latter allows the user to visualize real-time plots of sensors' data, and is visible in figure 7. The toolbar is also hideable by clicking on the bottom-left button *"BAR"*, in order to appreciate a full screen visual.

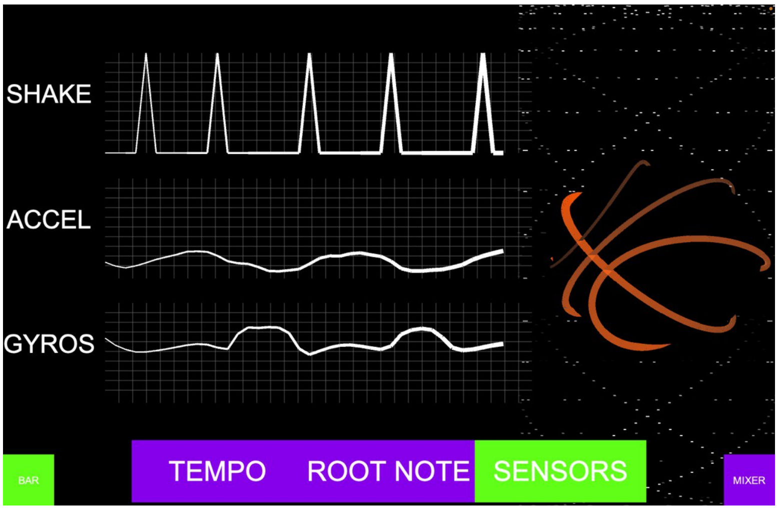

Figure 7: GUI Sensors Window

A mixer window is also available and can be accessed by pressing the dedicated button in the bottom-right corner. In this window, visible in figure 8, the volumes of the different instruments can be adjusted using the faders. The values in dB are adjusted in Supercollider to match the amp values used there. Mute and solo buttons are also implemented for each channel. The master volume can be adjusted either through the GUI or through the potentiometer placed on the Arduino case. The master volume also modifies the brightness parameter of the basketball's lines.

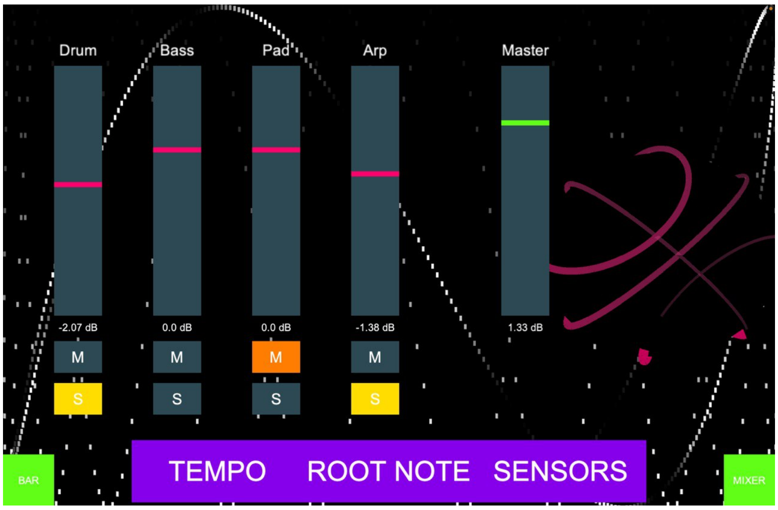

Figure 8: GUI Mixer Window

Conclusions

The subject matter allowed the experimentation of different ways to implement a Computer Music System, using for this purpose many different tools and environments such as sensors, Arduino, Processing and more. The result is a system that produces music following the movements of a basketball player, and can provide a pleasant background for a training session. The system has been tested successfully in an actual training session, and proved itself to be pleasant and encouraging for the player. However, as seen in the following sections, many improvements are possible.

Future Development

The system can be further improved, as it has some flaws, such as needing the player to wear cumbersome equipment. Furthermore, because of the limitations of the hardware (of not being able to discriminate with small ranges of values), the amount of ways the music can be generated by combining the different states is small, and can end up sounding limited and dull. Therefore, future developments might involve using different monitoring devices, that are different both in quality and in nature compared to the present ones. By using more devices, smaller ones, or more precise ones, the system could in fact offer a larger variety of musical accompaniment, as well as more controlling features, while also being more comfortable to the user.