# Computer Music - Languages and Systems
# Homework 3

*Juan Camilo Albarracín, Michele Murciano, Emanuele Pavese, Camilo Trujillo, Francesco Zese*

## 1. INTRODUCTION

Grasping the fundamental concepts regarding computer music languages, including different pieces of software and the possibilities they offer in terms of sound synthesis and processing, certainly opens a wide spectrum of prospective implementation areas and promising projects to be developed. Nevertheless, embarking on a rather complex endeavor with only one of these tools at hand can result in a very limited outcome. From there the vast importance of Interaction Design and Communication Protocols. Adding these tools to the belt allows the developer to take advantage of the most useful features of each building block of the project under construction, generating dynamic interactions between them and coming up with an integrated solution for the problem at hand, fulfilling the outlined objectives in the most efficient way possible.

With regard to this document, the design of a complete and original music system is to be achieved. The primal idea parted from the intention of developing a re-interpretation of a traditional music instrument while showcasing to a high level all the capabilities put at disposal by the interaction of the selected tools and protocols, which would be accomplished by structuring a music performance system with interactive components. The selected instrument for this case was the harp. Having reached this point, the selection process for the tools to be used was at hand.

In terms of hardware, the desired development was an interaction system unit that could capture gestural information similar to the natural hand motion of a harp player when interacting with the strings. Regarding the other computer music unit, a synthesizer that could emulate the sound of a traditional harp was to be designed, at least as a parting point before the implementation of other effects and elements for adding complexity to the sound. Similarly, the graphical component of the project was decided to be solved with a feedback interface simulating the behavior of the harp strings when getting plucked, of course, adding to that idea other visual components responding to the information captured from the other system's constitutive blocks. Lastly, the communication protocols that would allow this development were to be defined and developed.

Specifics about the definition of each of the formulated nodes for the development of the project, as well as the implementation phase and results, are presented throughout the present document.

## 2. OBJECTIVES

- Designing a complete music system featuring interaction design principles for gestural information capture and audiovisual feedback.
- Implementing an interaction unit for capturing user's input and sending MIDI messages over USB to communicate with the computer music unit.
- Developing a computer music unit for emulating the harp sound with responsive features to MIDI messages, such as Note On, Note Off, Pitch Bend and Control Change.
- Structuring a visual interface, resembling the strings of a harp, that reacts to incoming messages from the computer music unit.

## 3. IMPLEMENTATION

### Interaction unit with Arduino

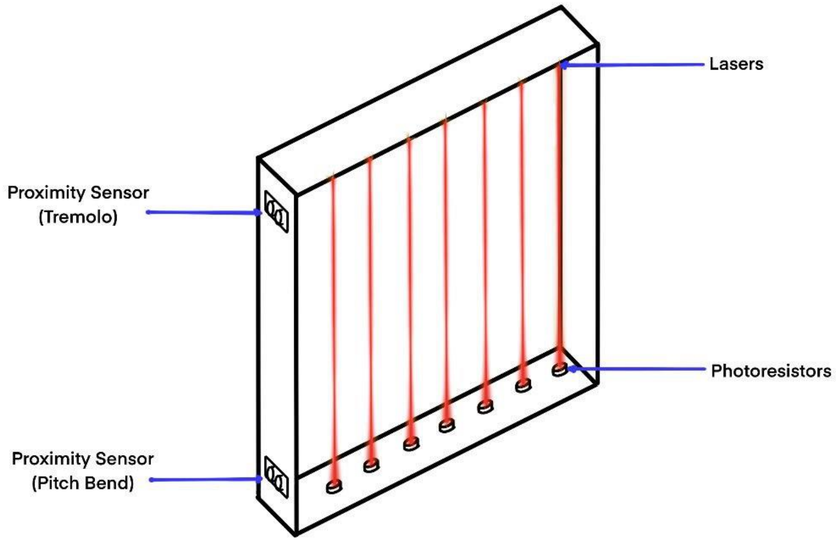


*Figure 1. Interactive unit sketch*

To capture the user gestures, it was decided to build an interaction unit that resembles the shape of a harp. To do so, a set 7 of seven lasers were used to emulate the strings and the photoresistors were aligned opposite to them (figure 1). In this way, the user interacts with the unit by interrupting each laser beam, this causes a decrease of the light on the photoresistor hence a drop of the electrical tension after it, this last is interpreted by Arduino as a Note On message. Similarly, when the beam reaches the photoresistor again, that causes a Note Off to fire. The changes in the tension are measured by an analog Arduino pin, which is connected after the photoresistor, as can be seen in figure 2.

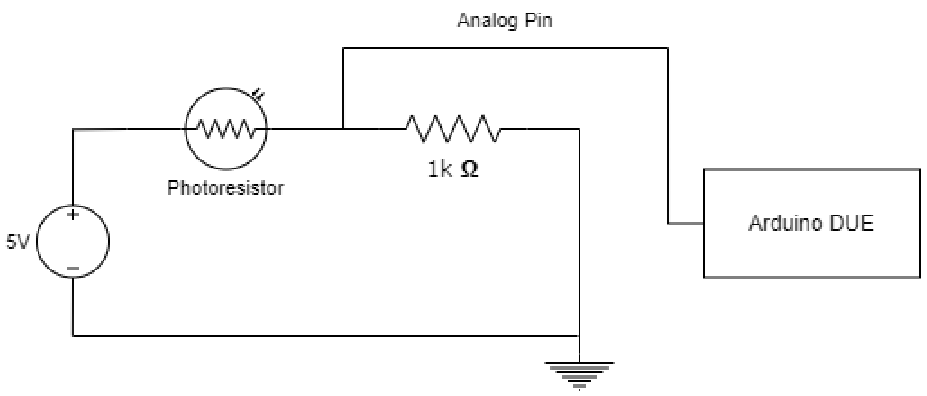


*Figure 2. Circuit schematic for each photoresistor.*

This interaction unit also features the possibility of modifying the pitch (+/- 1 semitone) through a proximity sensor. When performing, the user can use its hand to send pitch bend messages, the closer the hand to the sensor, the higher the pitch and vice versa. In a similar way, using a second proximity sensor, the user can fire control change messages which, in the music unit (next section), apply an amplitude modulation effect (tremolo). It is important to address that the distances readed by the sensors were restricted to 2 cm and up to 20 cm, so those distance values are mapped into a range from 0 to 16383 for the pitch bend, and between 0 and 127 for the control change.

Another comment on the Arduino development, for this project was implemented the library MIDIUSB which extends the environment giving to supported boards (as Arduino DUE, the one used in this project) capabilities to act as a MIDI instrument over USB. Taking into account what was described in this section, the following diagram depicts the general flow for note on - note off interaction:

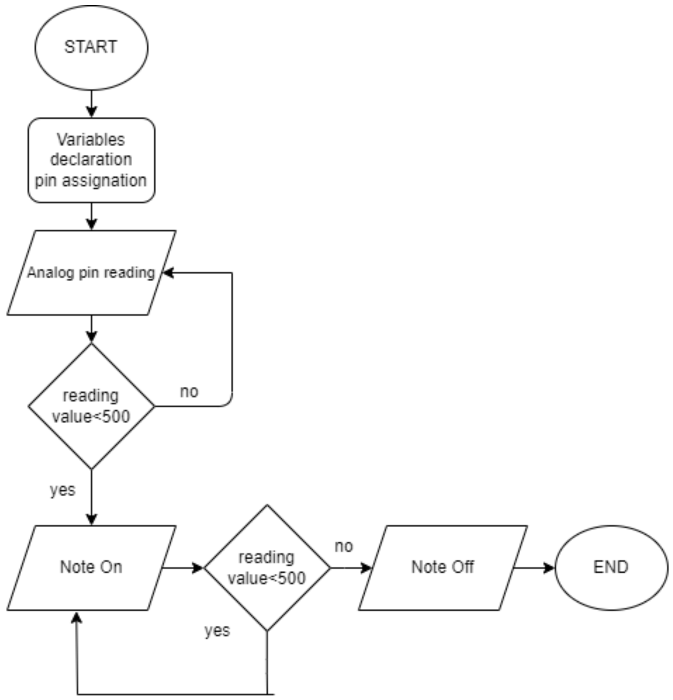


*Figure 3. Note On/Off diagram*

**Music Unit in Supercollider**

As for the sound, since the instrument is inspired by a harp, a sound that was as close as possible to that of a plucked string was chosen. In order to get this kind of sound, the *Pluck* class was used in the *SynthDef*..

```
Pluck.ar(in: 0.0, trig: 1.0, maxdelaytime: 0.2, delaytime: 0.2,
    decaytime: 1.0, coef: 0.5, mul: 1.0, add: 0.0)
```

*Figura 4. Pluck class in Supercollider*

Where as *in* (excitation signal) was chosen *PinkNoise* and as *trig* an *Impulse*.
In order to let SuperCollider understand all the midi messages sent from the arduino, different *MIDIdef* were implemented. *NoteOn* and *NoteOff* are the fundamental *MIDIdef*s. They receive a midi message containing velocity and note number as parameters. The velocity is fixed at its maximum value (127). The note number specifies which note is played, since arduino is capable of sending midi messages, for each photoresistor it is assigned a midi message with the right midi note number, starting from C4 (60), up to B4 (71). Each note number is then converted into a frequency value using the method *midicps* that converts the note number into cycles per seconds.

Other two *MIDIdef*s were implemented, one to control pitch bend and a second one to control the tremolo effect.
*MIDIdef.bend* receives a pitch bend message that is 14 bit long it goes from 0 to 16383; in order to have sharps and flats this value is mapped to the interval [-1, 0, 1] it is 8192 when no pitch bend is applied, 0 when a flat note is played, 16383 when a sharp note is played. The *bend* value is initialized at 8192 so the notes played are not pitched in either ways. The bend parameter is mapped in a linear fashion using the *linlin* message, doing so it does not have discrete values but continuous values.

```
synth.set(\bend, val.linlin(0, 16383,-1,1))};
```

*Figure 5. Assigning pitch bend value to the synth*

When a note is bent it will not go from natural to sharp/flat immediately but it will go through all the intermediate values between the natural note and the sharp/flat one. It works like the pitch bend wheel found in synthesizers. Having control over the pitch bend parameter allows the user to be more expressive because it can be used like a vibrato effect.

As mentioned in the previous section, Arduino also sends Control Change messages. In the case of the project the CC message is used to control the tremolo effect, in particular the frequency of the tremolo effect. Tremolo is a modulation applied to the amplitude of a signal. To achieve this effect the sound generated in the *SynthDef* is multiplied by a sinusoid whose frequency is controlled by CC messages sent from the sensor to the arduino and finally to SuperCollider.

```
modulator = SinOsc.ar(trem).range(0, 0.8);
snd = snd * modulator * amp;
Out.ar(0, [snd, snd]);
```

*Figure 6. Amplitude modulation (Tremolo)*

Since the tremolo effect is controlled by a CC message, a *MIDIdef.cc* is needed. The CC message sent from the Arduino is composed of three arguments: channel, control and the value (from 0 to 127) read from the sensor. With respect to the control, a specific choice is needed in order to avoid SuperCollider to pick up messages from all the available controls, as far as the project is concerned control 10 was chosen. The *trem* parameter is then mapped in a linear fashion in the same way as the *bend* parameter.

**Graphical feedback unit**

In this complete Computer Music System, the graphical feedback unit plays a fundamental role. Indeed, given the low luminous intensity of the light beam emitted by the laser, the strings of the harp are not clearly visible. Consequently, the main idea on which this graphical feedback unit is based is to provide the observer with a visual reaction of the movement of the string being plucked at that moment.

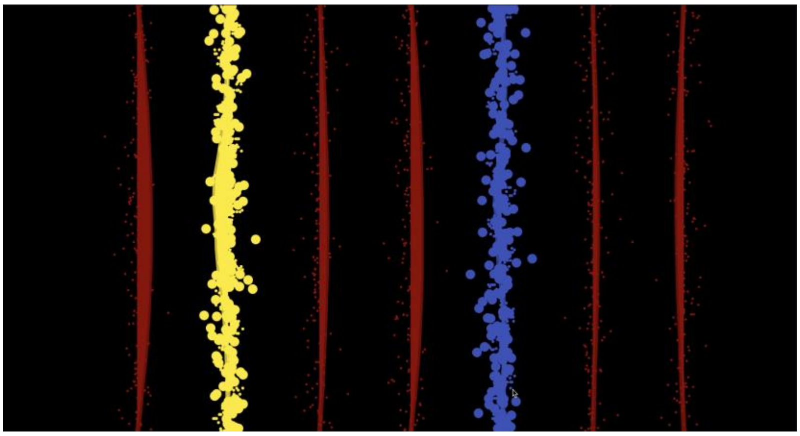

*Figure 7. Graphical feedback on Processing*

In order to make the perception of such visual reaction even more immediate, it was decided to project the output of the graphical feedback unit onto a cloth affixed to the back of the instrument. In this way also the instrument player has a better perception of the string.

Concerning the graphical feedback unit implementation, the approach on which it is based is the following:

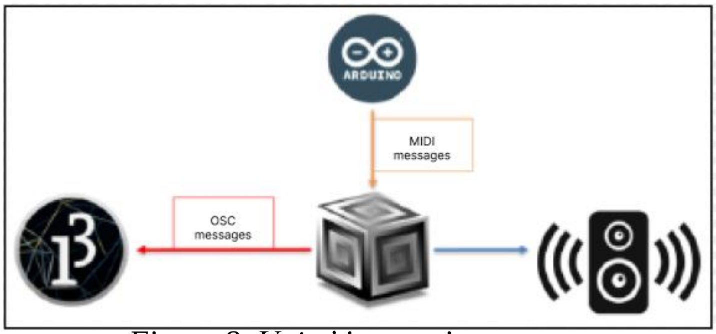


*Figure 8. Units' interaction*

As it can be seen from the figure above, the communication between Supercollider and Processing is made possible thanks to the exchange of OSC messages.

In addition, to provide the observer with a visual reaction to the pitch bend and tremolo effects, it was decided to implement two other visual effects:

- the first one is the change of color of the string according to the pitch bend.
- the second one is the change in size of the points that surround the string according to tremolo frequency: the higher the frequency, the larger the size of the points.

## 4. RESULTS

As stated in the objectives, the final product is a system in which the hardware, the sound synthesis and the visual feedback interact with each other in a functional and smooth way. The use of the MIDI protocol (for the notes and for the pitch bend) allows using the instrument not only with Supercollider, but also with any Digital Audio Workstation.

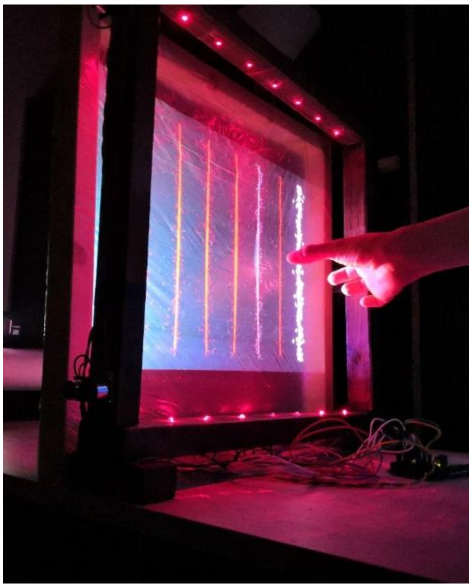

*Figure 9. Nah: Computer Music System*

## 5. CONCLUSIONS

- It was designed a complete music system, that captures the musician gestures and achieves to give audiovisual feedback. This was done through the implementation of different units that communicate between each other using protocols, such as MIDI and OSC.
- Using lasers, photoresistors and proximity sensors was accomplished to build an interaction unit in that resembles the shape of a harp.
- Using the supercollider environment, it was able to synthesize the sound of a plucked string that is responsive to midi note On/Off messages and through pitch bend and control change messages it was possible to control two different effects (pitch bend and tremolo).
- Via OSC protocol, it was developed a graphical interface that emulates the harp strings in a complete different fashion, responsive to the musician/user gestures and the incoming messages from the music unit.

## 6. FURTHER DEVELOPMENTS

- Giving the user the possibility to control the note velocity would add more expressivity and possibilities to the music system.
- The initial idea was to use a proximity sensor for each string, allowing to perform different pitch bend on different notes, and above all allowing to play altered notes more easily. Unluckily the ultrasonic proximity sensors cannot be used near to each other, because of the interferences between them. This feature can be implemented in the future using different proximity sensors, like IR proximity sensors.
- Furthermore, more strings can be added to include altered notes, taking into account this directly affects the measures of the interaction unit.