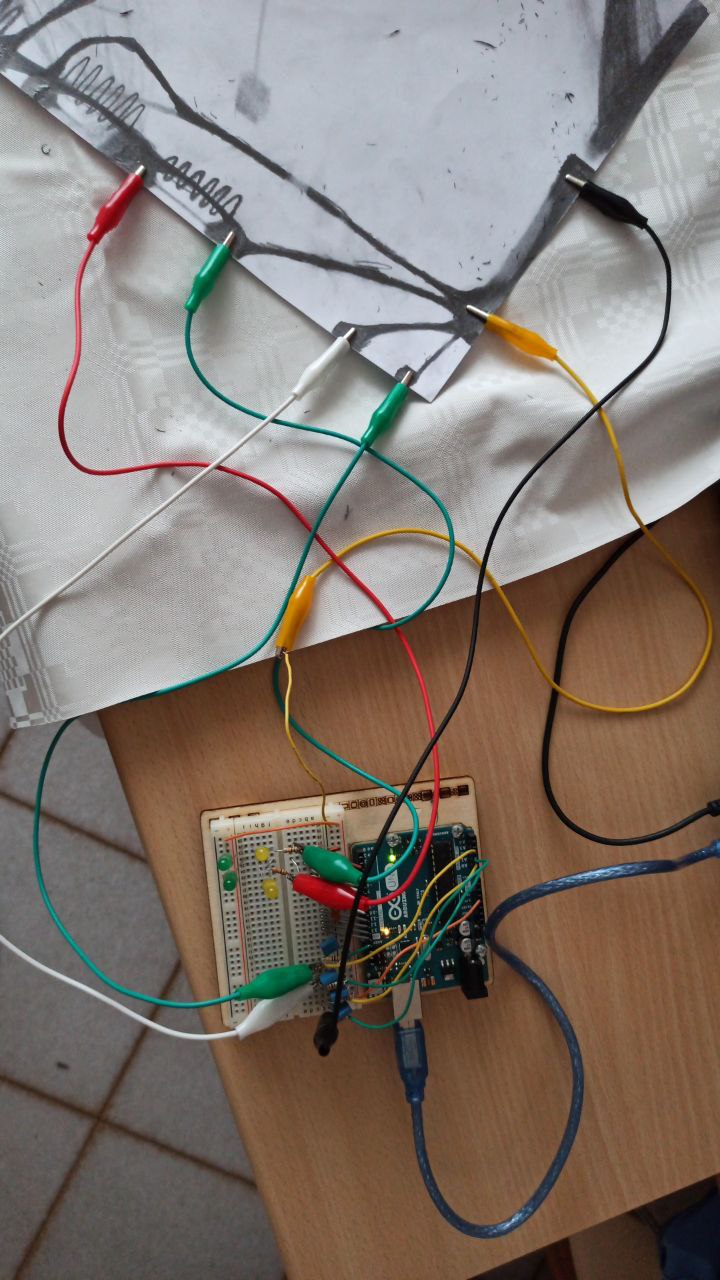
**Nonlinear Quasistatic Quintet**

Graphitizor



Group members: Marco Cristofalo, Marco Berasconi, Francesco Barraco, Bram Steenbergen, Geoffrey Horn

30/05/2022

Department of Music and Acoustic Engineering

Politecnico di Milano

The group known by the moniker of Nonlinear Quasistatic Quintet (NQQ) has been tasked with the creation of a complete computer music system with interaction design principles. In essence, the group was given a variety of software programs and hardware systems to create a system in which the user can create music/sound utilizing a unique system of creating the notes. The goal of the project is to be a culmination of the previous two projects, in which we utilized both Juce and Supercollider. However, this project takes the previous projects a step further by introducing new software programs and interactive hardware.

For this project, the new software that was introduced included Processing for user interface feedback, Bela and Arduino for interaction with the user, and other programs that were optional depending on the direction that the project was to be directed. Groups were not limited to only these options, but if a group wanted to use a phone to provide user input, for example, then they would be required to learn the manner in which the input device delivers said signals to the Juce and Supercollider programs.

In addition to learning new software, the class was also introduced to Raspberry Pi and Arduino hardware. Arduino is one of the most common user interface tools for engineering students, as the boards are cheap and easy to use, and they support a large variety of sensors to allow a wide array of inputs.

Additionally, there is a robust online community of Arduino users, so there are lots of resources on the internet for Arduino projects. All members of NQQ had utilized Arduino boards in prior engineering courses and were enthused to be operating within the framework once again. It is for these reasons that the group made the decision to work with Arduino.

The next step was to come up with an idea for the project. The group put pencil to paper to begin drawing out ideas and prototypes and creating lists of other creations from YouTube and Reddit, essentially brainstorming for any sort of inspiration. As writing and doodling continued, an idea emerged to create music using the pencil and paper interface. Perhaps a pen that tracked the angle and pressure of the writing and transcribed this into a frequency and amplitude? Could paper with transducers be created to track the location of writing to then create a note? Additionally, could graphite be used to direct the flow of current through the paper so that what is being draw creates some sort of sound?

Of all these ideas, the one that interested the group the most was the concept of turning drawings into sound. However, this idea in itself was still very vague and needed to be focused. The first step was to test the feasibility of using graphite to carry signal from an Arduino. After some very rudimentary testing, it was discovered that the amount of signal could be modulated depending on the weight and length of the line. As pencil graphite has very low conductivity, drawing a line that is too long would not allow enough current to pass. Additionally, lines that were too thin also were not conducive enough for the Arduino board. However, within the confines of a sheet of paper, these is plenty of surface area to allow for a wide variety of shapes to be drawn while still ensuring that the lines supported the flow of electrons.

With the feasibility study accomplished, it was time to narrow the scope of the project to find what the paper would be used for. Essentially, what would the paper “instrument” allow the user to do. Alas, the modularity of the paper user interface allows for endless possibilities. However, there were two logical directions for the group to take.

The first option was to create a sort of “beat pad”. In this configuration, the user would be given a series of drum noises, connected to specific Arduino channels, allowing the user to draw the nodes to look any way that they desired. Upon connecting the nodes, either using the fingers of the user or by drawing lines, said percussive synth would be played.

The second interactive medium was a sort of MIDI looper, in which the user could connect certain nodes to create a loop of a specified sound, as in a bass tone. Connecting certain other nodes with pencil lines would activate the additional synth notes to be played, allowing for a series of arpeggiations to be played. The amplitude of the note could be altered depending on the length and thickness of the line connecting the nodes.

Ultimately, it was decided that both modalities could be integrated into the project, and it would be within the Processing user interface that the musician could select the type of input setup that the user would be drawing. In fact, the implementation of a user interface that gave the user the option to choose which style was being drawn would be most optimal. This would allow the user to choose the sound that the nodes produced, allowing for be the optimal setup.

Once the initial brainstorming was complete, the time had come to begin the creation of the actual system. The first step was to build the physical system utilizing the Arduino board, resistors, connector cables, and paper. Figures 1 and 2 display overall wiring setup, along with a demonstration of how the Arduino board connects to the paper. Figure 1 shows a paper interface with two digital and three analog connections.

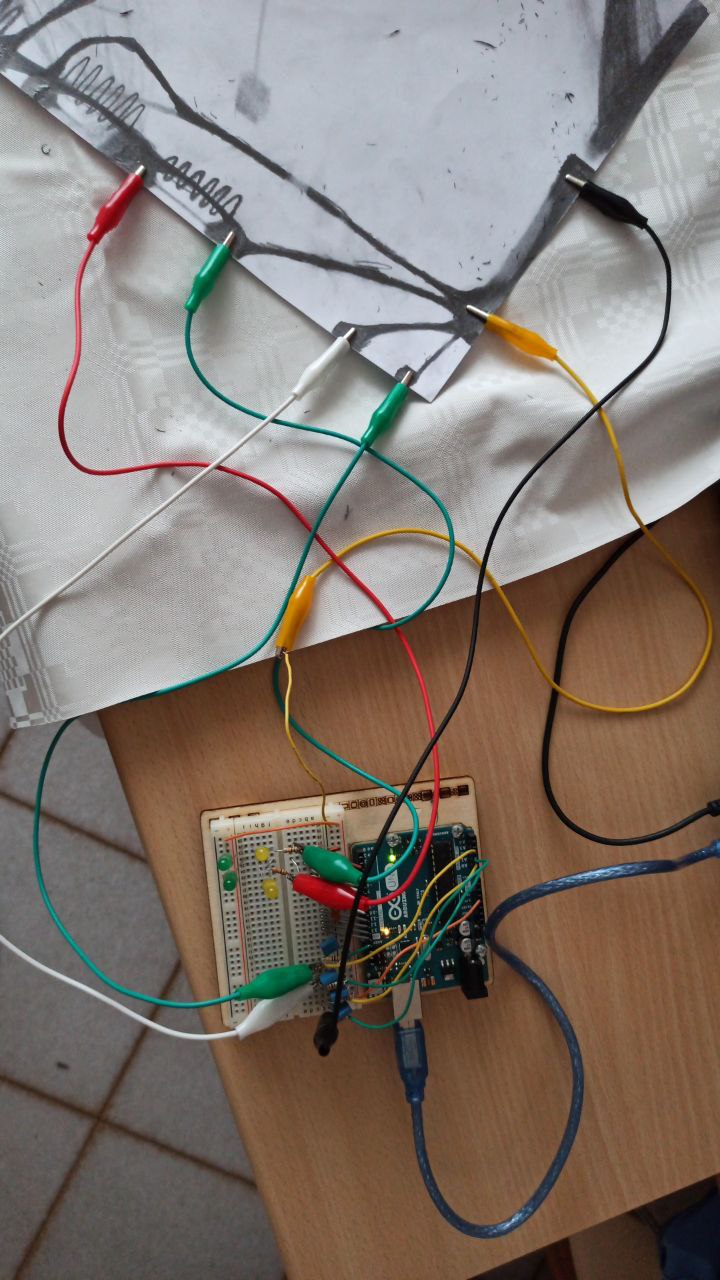
****

Figure 1: Physical interface

**A picture containing electronics

Description automatically generated**

Figure 2: Arduino board layout

The Arduino board supports 14 digital pins and 6 analog pins (connected to ADCs) which can be used as digital. However, pins 0 and 1 are reserved for serial transmission and can’t be used in this configuration, leaving us with 12 digital inputs. With these limitations, it was decided to use the 12 digital inputs as control signals for a 12-tone chromatic arpeggiator, in which a LOW value means that the selected note will be played in the arpeggio loop.

The digital pins are connected with an input pull-up configuration, where the graphite path going from GND to the input is the interrupter, as shown in Figure 3. The problem with this configuration is that the interrupter is supposed to be very low resistance while the graphite path resistance is in the order of MΩ (41kΩ is the bare minimum with a less than one centimeter path length), hence the pullup resistor’s value must be at least 10MΩ for the circuit to work properly. Due to low material availability, a diode pull-up configuration with a parallel condenser for stabilization was implemented instead, as shown in Figure 4. Two LEDs in series were used in order to achieve a 3.9V voltage drop.

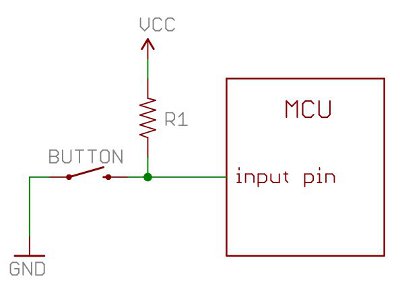


Figure 3: Input pin wiring configuration

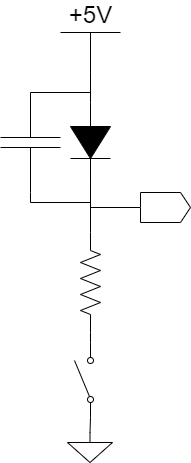


Figure 4: Diode pull-up configuration wiring

This circuit is far from optimal and, for example, is still subject to output variation due to EMC parasitic voltages. To overcome this problem it is beneficial to use a battery powered source instead of AC current.

From the microcontroller part, the digital input acquisition is done with a simple polling mechanism inside the Arduino loop. To increase stability, a “wait and see” type software control section was implemented during every pin reading. Therefore, the reading is done twice after a fixed wait time, and the software will proceed with further related operations only if the two readings are equal.

Analog inputs work in a similar way, with the electrical configuration being a voltage divider made using a 10 MΩ resistor and a 10 uF parallel capacitor for stability, as shown in Figure 5. The microcontroller uses its standard ADC reading function followed again by a “wait and see” control section.

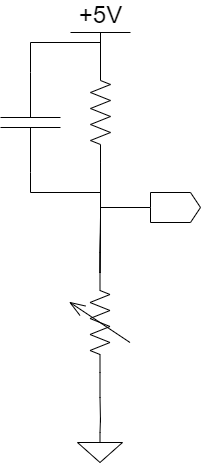


Figure 5: Analog input wiring

After the physical user interface was created and the Arduino wiring sorted, the Supercollider portion was the logical next step to implement, as this was the portion of the project that would hold the sounds to be played. While the Supercollider code was created and compiled entirely by NQQ, many of the synths that were used were pulled from either online resources or previously created synthesizers. For example, utilizing online repositories for a kick drum sound was significantly more efficient than spending hours of coding building a new synth and tinkering with all the parameters. However, this practice is standard in the world of programming, and saving time by not creating the synths allowed the group to focus their effort on the implementation of the sounds.

Therefore, within Supercollider, it was decided to implement a system of partitures capable of producing an arpeggiator and a drum set based on the inputs from Arduino and Processing. Additionally, the messages from Processing would allow the user to decide from a wide variety of sounds, with Processing being the interface between Supercollider and the physical paper system. Within the Supercollider program, a total of six tonal synths and four percussion synths exists for the user to implement. The tonal synths include saw bass, violin, Rhodes synth, pad, pluck, and a sinusoidal wave synth. The four percussion synths include kick, snare, hi hat, and clap. These synths sound as close to the real instrument as could be achieved within our Supercollider programming capabilities. However, the decision to choose to utilize a “violin” synths as opposed to a true violin is a creative decision by the user and will therefore be more than sufficient for the sound that a user of the program wishes to achieve.

The final arpeggiator is composed of three different arpeggiators; the first operates at low frequencies (transposed of -24 by the central C of the keyboard), one at medium frequencies (octave of central C), and one at high (transposed of +12 by the central C). These three arpeggiators have three different sounds which are selected from the Processing GUI. The duration of the single notes of these 3 arpeggiators are set as 2 for high, 4 for medium, and 8 for the low, so they create different polymetric structures based on the number of notes switched on physically on the paper.

There are a multitude of inputs from Arduino which interact with the Supercollider program. There are 12 digital signals for the semitones of the whole chromatic scale, which build the sequence of the arpeggiator. The order that these notes are played is determined by the order in which the user links the graphite pins on the paper, allowing the user to determine the look of the paper interface. There are three analog signals, which are continuous signals representing the tensions between the pins in order to modify the volume of the three arpeggiators. Another analog signal modifies the pitch by changing the starting note of the chromatic scale. A third analog signal determines the overall volume of the drum set in the same manner that the overall volume of the arpeggiator is determined. A fourth analog signal modifies the beats per minute of the arpeggiator and drum set functions. Altogether, all of these inputs allow for a simple physical setup that is easy for all users to set up on any piece of paper.

The communication protocol between Arduino and Supercollider was initially thought of as a serial MIDI-like, but it has been eventually slimmed down since the velocity byte is unused in this case. Its final form is:

**[COMMAND] + ‘a’ + [VALUE] + ‘b’**

As in MIDI, the first four bits of the COMMAND byte identify the type of command (mainly NOTE ON and NOTE OFF), while the last four select the channel. The first channel is the DIGITAL one, which is comprehensive of all the digital pins, while the analog pins have a dedicated channel one for each. The VALUE byte identifies the pin number in the case of a digital reading or the voltage reading in the case of an analog one. Characters ‘a’ and ‘b’ are just dividers to let Supercollider know when a message is finished.

Serial messages are delivered only when a new change in the configuration is detected, meaning that for digital pins they are triggered by on-off transition (drawing a line or erasing it). Analog pins are triggered by changes in the voltage reading. Messages coming from analog inputs are triggered only after a certain voltage variation threshold. This is done to limit the stream of analog data since voltage readings across a graphite path are highly fluctuating. Additionally, the analog reading has a minimum threshold below which the graphite path is considered completely open and a NOTE OFF message is delivered.

For the digital user interface, it was necessary to represent the paper sheet in Processing to allow for a logical control of all the sounds that the paper setup could produce. The user inputs from Processing include the pattern of the drum made by the sequencer, the volume of the individual sound of the drum set, the sound of the arpeggiator, and the duration of the notes for the arpeggiator. Ultimately, the most logical way to represent the two parts of the project was to separate the GUI into two screens: the Assignment view and the Sequencer view, as shown in Figure 5. The user can switch between the two screens using the “←” and “→” keys as directed on the screens.

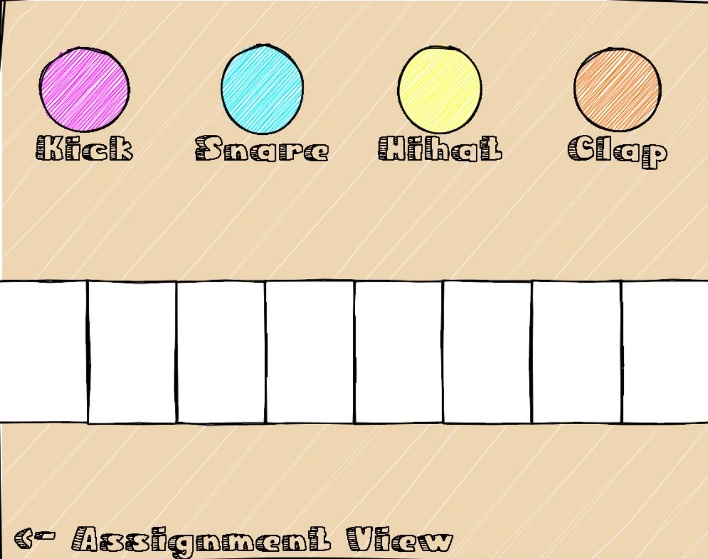
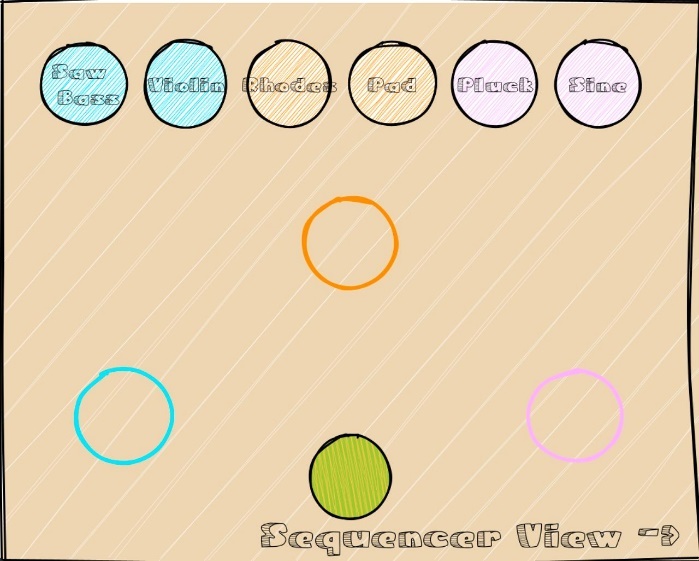


Figure 5: Sequencer View and Assignment View screens side by side

The Assignment view allows the user to drag the pads which represent the instrument sound into the empty pads to send OSC messages to Supercollider. This input tells Supercollider what instrument to play in which frequency range. There three pads to represent the three ranges: low frequency, medium, and high. Each of the pads is color coded, and the instruments are also color coded according to the range that the programmers have determined at which they sound best. To allow for maximal creative flexibility, all pads can be connected the whichever instrument the user desires, as the color coding is only a suggestion.

The Sequencer view presents the user with four different colors and draws heavily from the illustration theme of the project. Each of the four colors is labelled with a specific drum sound. By clicking on the color, the user can then paint within the pads located below. The 1 through 5 keys can be used to choose between five brush sizes for fine-tuned illustration. The amount of color within the specified pads tells Supercollider how loud the instrument is to be played. Therefore, by filling in the various pads, the user can “mix” the drum set to their specifications in a highly interactive interface. This is a fresh take on programming drums on a computer. In addition to choosing which drums are being played and at which amplitudes, the number of beats can be altered between 1 and 8 using the + and – keys to either add or subtract a beat from the sequencer. The “e” key activates an eraser to undo illustration within the pads, and the “r” key resets the whole sequencer to a blank slate.

While the project functions exactly as designed, there is certainly more to be desired that could not be accomplished due to time constraints, which the group discussed. First, the limited availability of instrument and drum kit sounds could be improved. Six instruments and four drums do not allow for a wide variety of sounds for a potential user of the program, and this could easily be built out with more development time.

Another hurdle that this project encounters is a base level of knowledge of how the paper needs to be drawn and the nodes attached to the paper. Perhaps allowing the user to draw the design on the user interface as well, and then assigning the nodes on the GUI would make it less confusing to figure out which notes are being used for the arpeggiator, but this would require a complete overhaul of the GUI, or the addition of a third screen in Processing.

Additionally, as it stands now, the sequencing is very specific, and the nodes need to be attached in certain orders, limiting the creative choices. Moreover, the limitations of graphite as a conductor means that the user needs to know exactly how the lines need to be drawn, including length and line weight characteristics, so if there are graphites that are more conductive, this would remove these limitations.

NQQ also discussed introducing a beat pad that could be activated using the fingers as conductors. However, this did not fit the drawing theme, but it would give the user another option for creating their drum portion of their project.

As the prior paragraphs show, there are many directions that this vision could be taken. This project can be viewed as a proof of concept for what was created. However, it excels in demonstrating the capabilities of a highly modular computer music system utilizing easily accessible materials. All that is needed to begin making music are the connections for the nodes, a piece of paper, a pencil, and the program.

Ultimately, this project merges art and music in a highly visual and uniquely interactive manner. Both in the physical interface and in the GUI, the user is encouraged to utilize their artistic abilities in an amusing manner to create a new synthesizer every time the program is utilized. The possibilities are infinite for how the user decides to interact with this computer music system. In a drab world in which all synthesizers look the same, this is a breath of fresh air and a new take on music for all ages.