dWill Luna
Spring 2017
Electronic Music Instrument Design

Tonnetz Board Final Report
Will, Ronna, Nathan & Eric

Introduction

Our team spent the second half of the semester improving the Tonnetz Board that we built for our initial project. Despite some design alterations, our emphasis with version 2.0 lay in improving the construction of the instrument. We knew the proof of concept worked, and wanted a larger, sturdier build. In addition to doubling the size of the board from fifteen to thirty playable vertices, we also planned on installing a single LED underneath the center of each acrylic triangle.

Design

The border vertices of our first version were not very responsive to touch, as a portion of each rubber bumper was covered by the wood housing. To ameliorate this problem, we cut an additional layer of cropped triangles around the playable portion of the instrument face.

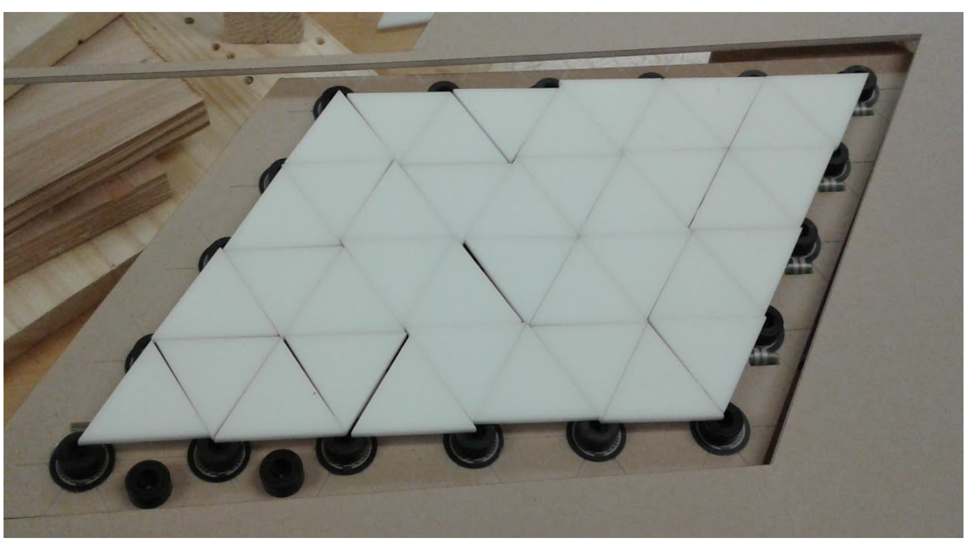


Figure 1. To make the surface more playable, a gap was cut between the playable vertices and the housing.

Since wiring the thirty vertices would now demand all but two of the analog pins of two Arduino Megas, we could only wire new controls that read discrete, digital outputs. Transposition had been difficult to control with a slider in our beta, so we mapped transposition to a twelve-way rotary encoder. This freed up the slider for global volume control.

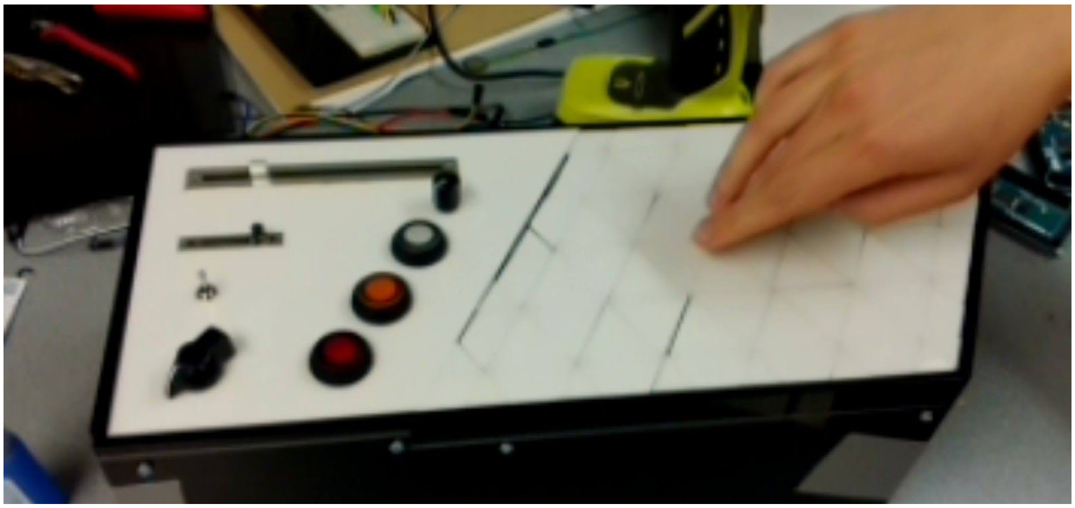


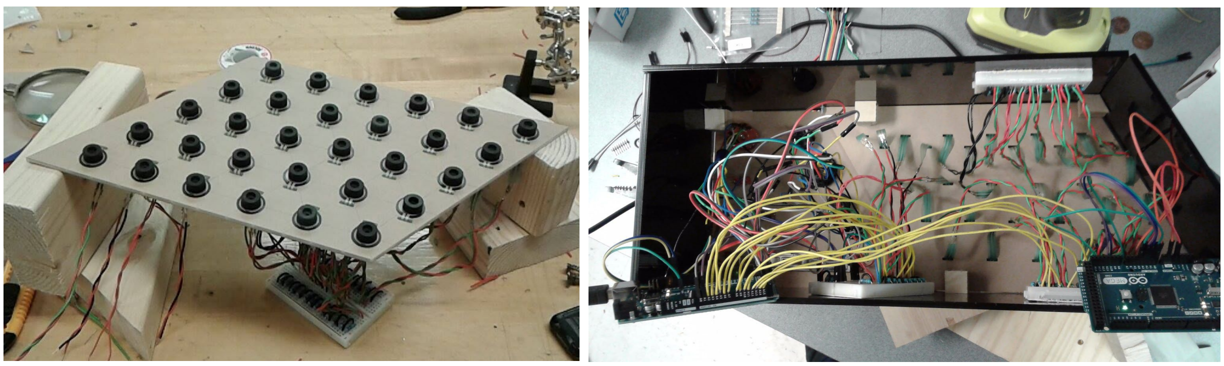
Figure 2. Two rotary encoders were added to the controls, for transposition and tempo.

Fabrication

The face of the board was recut to accommodate the two rotary encoders and the buffer zone between the playable surface and instrument edges. Small slits were also cut into the lower layer of acrylic, so the force sensing resistors could pass through.

The challenge of creating a sturdy build for our instrument was the two different layers of components - the top layer, which contains all of the acrylic, buttons, knobs and sliders, and the panelling which all of the resistors and bumpers rest upon. We used Ronna's suggestion of screwing wooden blocks into the sides of the housing, at variable heights. Drilling the holes into the blocks proved challenging, as the plywood split quite easily. The lack of a central support also meant that pressing down hard on the center of the Tonnetz board risks breaking the instrument. However, the build quality improved enormously.

The wiring was redesigned for neatness. Resistors and wires on the breadboard were cut as short as possible to compact the circuitry. Wiring was cut to the precise length necessary to reach a breadboard that lay centered and underneath the Tonnetz. In the final build, the breadboards adhered to the acrylic walls of the instrument. The wires connecting to the digital pins should also be cut to length in the future.



Figure(s) 3 & 4. Some forethought went a long way in cleaning up the wiring. However, the geographical realities of the housing and the Arduinos made the final build messier than intentioned.

Programming

Relatively little had to be changed in our Max or Arduino code for the second version of the instrument. However, we discovered that we had been incorrectly reading digital current, essentially running the circuit in the opposite intended direction. We now instantiate all digital pins with the digitalWrite(HIGH) command.

Nathan also coded LED-chord pairings, the former lighting up when the latter is pressed or played back in a recorded loop. This required using pulse width modulation to give each LED a variable brightness that corresponds to the aggregate note velocity of the three notes in the chord.

Audio

We decided to ditch Max altogether and implement audio playback with VST plugins. This was done because the architecture that allowed pseudo-MPE (Multidimensional Polyphonic Expression) demanded a separate Reason patch for every triangle vertex. Reason needs about one second to change each subtractor patch, meaning that changing every module-vertex pairing for our instrument would now take nearly thirty seconds.

The VST plugins allowed for these changes to occur instantaneously. However, this turned out to be prohibitively taxing on the CPU, and opening more than a single patch (thirty plugins) simultaneously routinely crashed the computer. VST and Reason both present unsolved obstacles to patch changes, but for different reasons.

Improvements

We did not implement everything in our design specifications, and future work on the instrument would be best spent finishing our original intentions. Recutting the wooden support blocks out of a material other than plywood would make the build much sturdier. A removable bottom could be attached with velcro to contain all of the wiring and improve portability. Using more than one color of acrylic would help distinguish between the buffer triangles and the playable vertices as well.

Handling patch changes with VST plugins without overloading the CPU remains an open-ended problem. The poly~ object in Max proved very difficult to implement, and there are no other clear solutions for improving the efficiency of communication with VST. However, implementing the recording and playback of loops, as well as an LFO, should be rather trivial.

We underestimated the time intensity of programming and wiring the LEDs. Having light patterns provide visual feedback to both live and recorded playback would drastically improve the usability of the instrument. A sturdier build that incorporates this feature could serve as a powerful tool in music education.