

James Hughes
April 10, 2008
Professor Lehrman

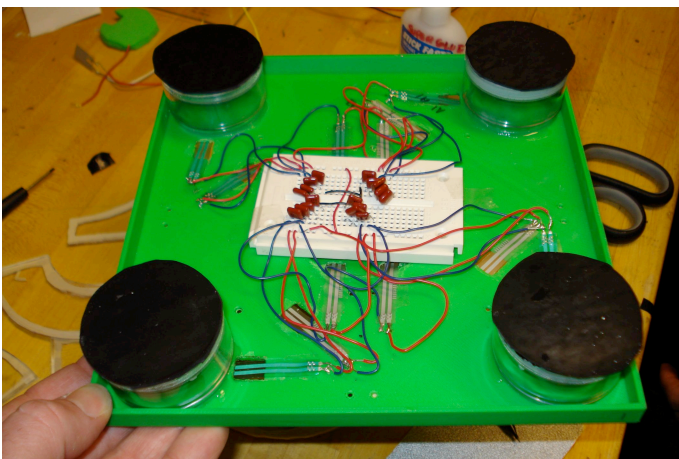
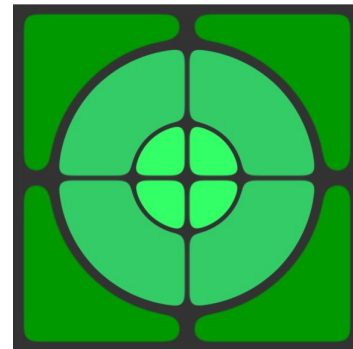
Xenbox

Goal

The artistic motivation for the Xenbox was to create an aleatory music generator, that is, an instrument where expression is left primarily to chance, not the user's input. To do this, we decided to create a square base of 12 buttons that would be placed at the bottom of a plexiglass column. The element of chance was left to bouncy "superballs" that were tossed into the column and then left to trigger pads at random as they bounced on the pads, each other, and the walls of the housing. The sounds created by the Xenbox are rich in harmonics and ever-changing, a perfect soundtrack to the path towards "Zen".

Construction / Wiring

The construction of the Xenbox was aided by our access to the Mechanical Engineering Department's 3D printer and laser cutter. The goals in construction were to make a durable, attractive, aesthetic chassis that would consistently trigger the max patch with most of its surface area. For the trigger pads, we decided to have two concentric circles inscribed on a square base, all of which is divided into four quadrants (right) allowing for 12 pads. The smaller center circle was slightly raised to allow for a different elevation and a lip for the balls to roll off of. We had hoped that the raised circle would be inches higher than the base, but the 3D printer could not accommodate that design. The 8" x 8" plastic base was designed in SolidWorks and was printed on the 3D printer and gave us guidelines for where to place the sensors, as well as drill holes for the wiring to allow everything to be hidden. The plexi-glass pad buttons were also designed in SolidWorks and were then cut by a laser cutter.



After the base was made, the sensors were soldered to wires and then attached to the base with their adhesive backing. We decided to use 12 FSRs – 4 large square FSRs for the outer pads, and 8 smaller, circular FSRs for the other 8 inner pads. We used FSRs in the pads because they allow for triggering by pressure over a flat surface and because they can be triggered even if only a portion of

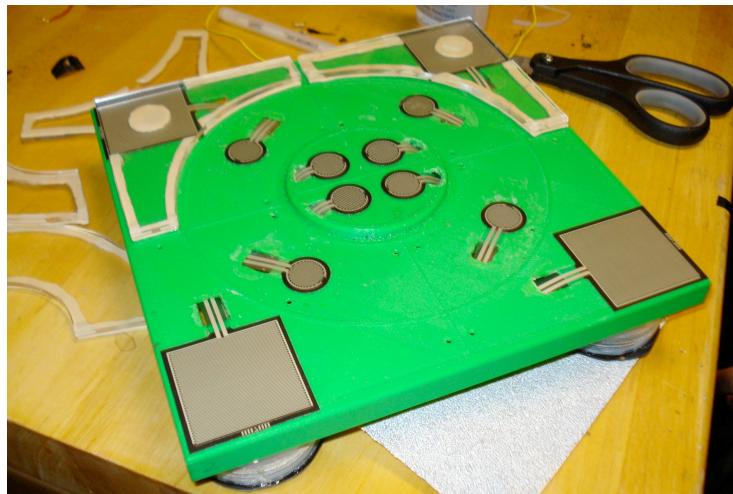
the sensor area is engaged.

After trying to wire the sensors through the drilled holes, we decided it was easier to make larger holes in the base and then feed the tails of the FSRs through those holes. This decision was made for aesthetics and so the wiring would not interfere with pressing down the pads.

After the sensors were in the place, the wiring to the breadboard was done. We tried to keep the wires organized on the bread board so that we could visually call up the different areas by quadrant. Each sensor went through a capacitor to filter out some of the noise, and this worked well in terms of triggering notes, but the capacitors also caused the FSRs to have a long release (up to 6 seconds). The breadboard was then wired with solid core wire to send out to the Doepfer box.

After the wiring was done, the box was flipped over and the buttons were attached. To ensure that the sensor would get pressed hard enough if the pad had pressure anywhere on it, we used thin rubber to cut out risers that rimmed the button on all but one edge as well as a small circle of rubber to trigger the FSRs. By leaving one edge rubberless, the pads had enough give to be triggered easily.

Our original plan was to have, in addition to all of this, 2 piezo sensors that would pick up the vibrations caused by the balls bouncing off of the housing, but the housing was not built until late in the process, and the piezo sensors could not be tested or incorporated. Although the 30" plexiglass housing came late in the construction process, it was made and allowed for us to bounce the balls freely on the pad without worrying about losing them.



MaxMSP / Reason

The foundation of the MaxMSP patch was an earlier patch that Renato had built called “The Steve Reich Generator”. The basic design of this patch is a series of synced, arpeggiated voices that sound on different division of the beat. Renato’s original patch was modified and improved to include control from the Xbox, more voices, and a new way to control a non-arpeggiated pad synth.

We used the “control ins” on the Doepfer box because the Max patch was based around make notes and was therefore control by bangs.

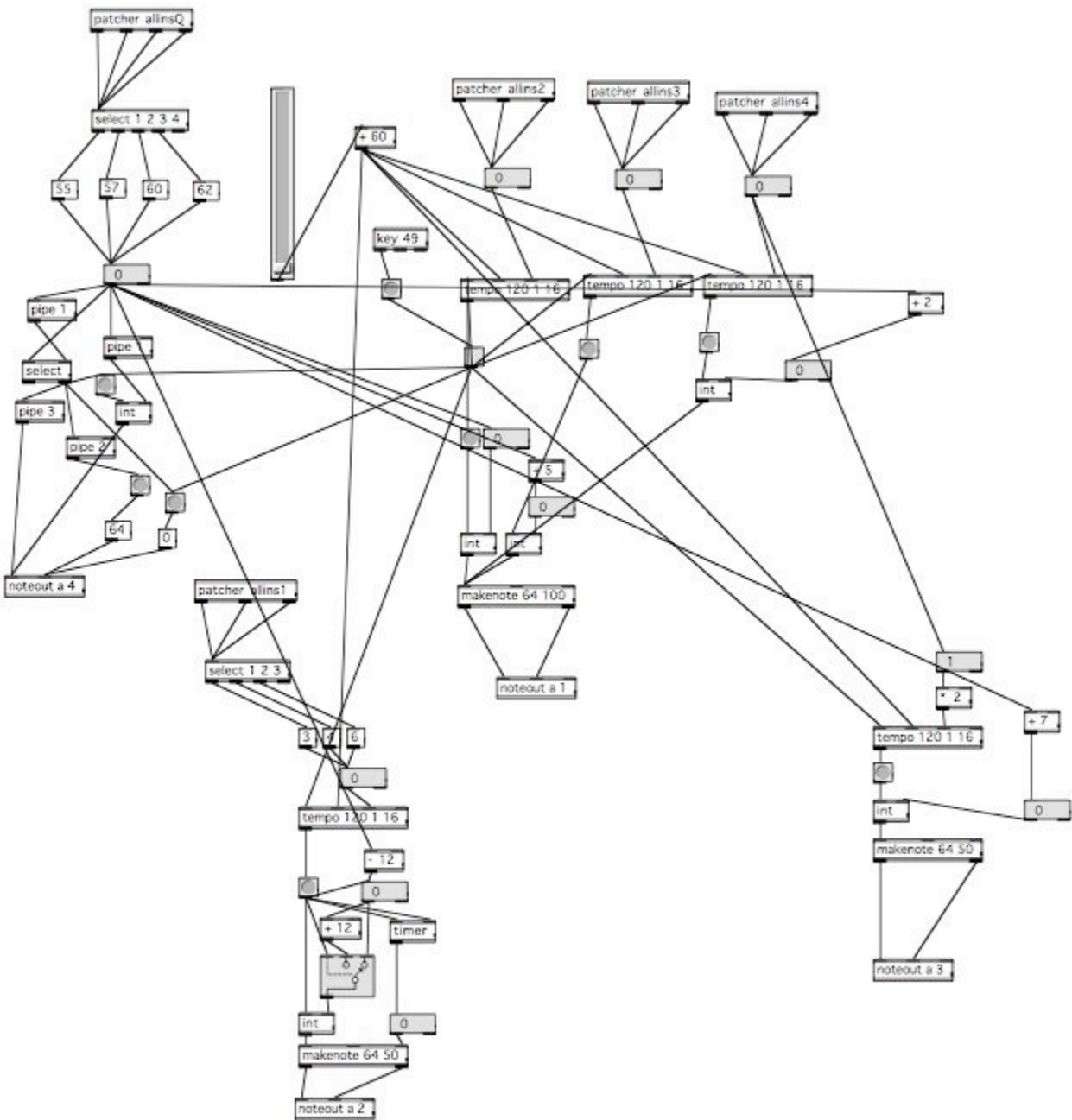
At the top of the patch are 3 patchers that determine the division of the arpeggiator beat. These are set up to work with the Xbox so that each of the twelve pads designates either a division of 3, 4, or 6. Beats are sent out at the designated division of sixteenth notes at the tempo set by the slider at the top.

Another patcher in the top left called “AllinsQ” serves to designate the root note of the generated music. The patch is set to the key of G, and is set up to play notes 55, 57, 60, and 62 (G, A, C, D) - the root, the ninth, the fourth, and the fifth.

At the left side of the patch are some pipes that allow the synthesizer pad to be played. The middle of the patch (switchable gate, tempo, and Makenotes) is the arpeggiator that triggers all of the notes.

In the “Allins” Patcher, we have some simple arithmetic that filters out the unwanted noise from the FSR signals. We made it so that only a certain range of pressures exerted on the FSRs would trigger notes or change parameters in the patch. This varied slightly from sensor to sensor, but, for the most part, the same ranges were used.

The Reason patch for this instrument was fairly simple, consisting of a line mixer, a few Subtractors, a Malstrom, and an NN19. The NN19 played a high arpeggiated vibes part, the Malstrom played the lead pad synth, and the Subtractors played the rest of the lower arpeggiated voices. The Malstrom patch had an interesting envelope on it so that it triggered an audible note at first, but if the note was held down, a filter would open slowly and new harmonics would sound. Each quadrant (three buttons) on the Xbox triggered a different scale degree. This scale degree is the note played by the Malstrom.



Summary

The biggest problem in this project was our timeline. The base was not constructed until late in the project, and from there on our progress was delayed. We wanted to test the sensors to be tested on the surface, so we did the sensor implementation in the Max patch after the FSR were attached. Additionally, we did not have the plexiglass box until the day the project was due making it impossible to attach piezos. The biggest lesson I learned here is that I need to schedule out the steps of my next project and allow for lots of time to accommodate the inevitable glitches in building an instrument.

Apart from poor planning, we did not run into that many glitches. One problem that limited our ability to effect different synth sound parameters was the noise created by having 12 sensors and all of our electronics concentrated on an 8" x 8" square. There was lots of noise between the sensors from crosstalk. Also, we had problems with the capacitors causing a very slow release on the FSR values. This could have been handled by special ordering capacitors more specific to the job, but it was too late at the point that we realized the problem.

Our group had a very aesthetic mindset when constructing the instrument. As three engineers, it was important that we had a good design and good constructability, and I think we achieved that. I'm especially pleased with the construction of the rubber-backed buttons, which work very smoothly and consistently.

All-in-all, I am proud of the Xenbox, but plan to schedule out my next instrument better, and incorporate more expression into it, as this instrument was a practice in chance.