Morgan Rivers
Storm Athletics: Tyler, Rachel, Morgan and Jackson
Final Project Report
5/12/16

The Glass Harp MIDI Controller

Introduction

The glass harp MIDI Controller is an expressive, instantly tuned, musically dynamic version of the traditional glass harp, an instrument which is commonly used by street performers and occasionally professional musicians. A glass harp is built with a series of glasses lined up on a table, and tuned with water, which damps the resonance on the side of the glass and lowers the pitch. The glass harp was an inspiration to us in the construction of our instrument.

Instrument Layout

The instrument we designed (Figure 1) uses rotary and linear softpots, sustain pedals, a trackball, a four-way knife and three buttons to detect player's gestures. Each of the softpots sends analog positional data corresponding to the location along the circle when pressure is applied. Thirteen of these are adhered across the top of the acrylic board surface. The large 18-inch linear softpot is the closest to the player and goes from vertically aligned underneath the center of the trackball to the right edge of the board. The softpots also send a nonzero value even at the very edge, so whether or not pressure is applied to the softpot can also be measured. The trackball sends out x-y values for velocity, corresponding to a spin through the x and y independent axes. This is placed closest to the player to the left of the softpots. The buttons send on/off, and the three used are placed above the softpot. The four-way knife switch sits the furthest from the player, and indicates which of the four positions its in through on/off signals.

On the floor below the instrument is a set of four pedals, each of which are pedals typically used for electronic pianos and which only have on/off switches. Three of these are mounted to a wooden board with epoxy. The last one is free standing like sustain pedals normally are, and is usually placed to the right of the pedal board.

Playing the Instrument

Playing the instrument involves the use of three modes; traditional, theremin, and sampled. These modes can be reached by using the first three values in the four-way knife switch.

Traditional mode

In this mode, the harp produces sounds emulating a real glass harp. The circular softpots can be imagined as glasses. Faster motion across the sensor allows louder notes to be produced, such that each softpot velocity produces its own loudness setting. Additional dynamic control can be achieved with the

use of the linear softpot. Moving this softpot allows the user to change the master volume, with a larger volume the further to the right the most recent press on the softpot was.

In this mode the trackball was intended as a means to replicate the sound of glass harp performers moving the table, thereby wobbling the water in the glasses. In real glass cups, this creates an oscillatory pitch bend. The glass harp does the same thing with the trackball, bending the pitch lower with a damped oscillation motion.

Additional pitch range is achieved with the octave switches operated by the pedal unit. Neutral octave is the center pedal, down an octave to the lower octave is the left pedal and up an octave is the right pedal. The sustain pedal is feature that's impossible to design with real glass harps, but sustains like a piano pedal normally would. The two leftmost buttons work as reverb increase or decrease, to allow the pitches to sustain longer and give a somewhat echo-like effect. Hitting the left-most button decreases reverb, hitting the middle button increases it. The right-most button is used as a transpose, allowing the performer to transpose the pitch of the piece played up from C to some other key. This is accomplished by holding the button down and hitting the softpot corresponding to the key desired.

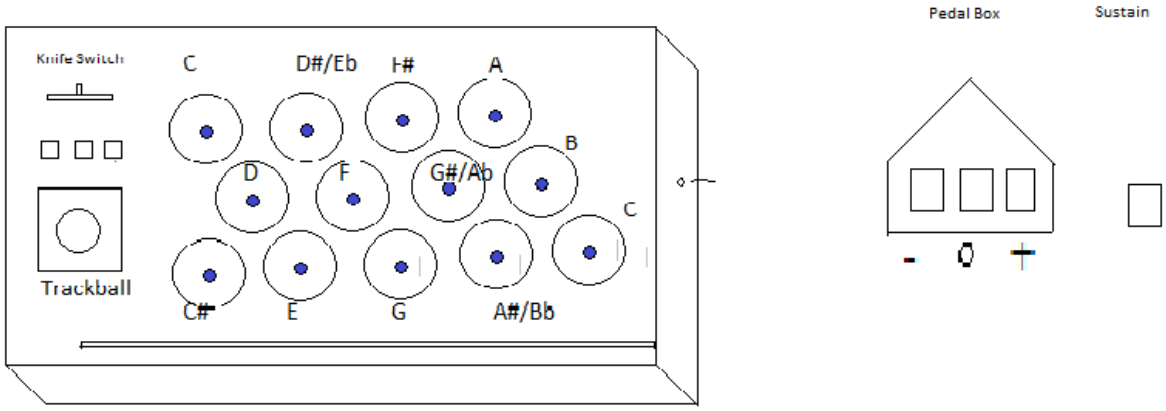

FIGURE 1: THE LAYOUT OF THE EHARP

Theremin mode

In this mode, the harp produces sounds emulating a theremin. As before, the octave range is controlled by the pedal, and the pitch is controlled by the choice of softpot played. However, in this mode the pitch is bent considerably by the position on the softpot. Placing one's finger on the vertical center of the softpot makes the note turn on, causing the theremin to produce the pitch corresponding to the softpot. Counterclockwise motion up to the top of the softpot will pitch bend upward, by approximately an octave, and by a similar amount by clockwise motion to the top. In this way vibrato can be produced by left-right swiping motion of the finger across the bottom of the softpot.

As before the linear softpot controls overall volume, in a way similar to the volume control with the left hand of the theremin. The three buttons produce exactly the same effect, except the left most button will turn off the note in addition to decreasing the reverb. The trackball and sustain are not used in this mode.

Sample mode

The final mode of the instrument introduces many new uses of the instrument components, taking advantage of the flexibility of sound-based instruments. The sample mode can use any set of tracks of a known tempo and time signature. The instrument uses max msp's sample playing ability to play these prerecorded samples forwards and backwards in time corresponding to the velocity at which the softpot is being rotated. Each softpot controls a different sample. When playing, the sustain pedal replaces its function with a sustenuto, and any track that is being played will be played at normal playback speed when the sustain is held. Holding down a note will produce an extremely slow playback, and as a result becomes low pitch and quiet.

True to the form of a sampling instrument, the buttons add noise and distortion to the track. Each button creates distortions on the quarter-note in 4/4, quarter note in ¾, or eighth note in the sample. The trackball creates similar distortions, but modifies based on the direction the ball is hit in the x or y axis.

Instrument Construction

Physical Construction

Overall the team adopted a programming first mentality, and reduced the number of moving parts, weighty components, and other artifacts of real instruments to a minimum. This allowed the construction of the instrument to be straightforward, though not altogether without its challenges. Initial construction involved the gluing together of wooden boards and the purchasing of a large sheet of black acrylic. Two holes were drilled into the wooden frame, and a jack for the sustain box was added, which was just a standard digital piano sustain pedal. The other hole allowed the placement of a usb jack that allowed easy connecting and disconnecting from the computer. Once the glue dried, the acrylic was cut in half (by handsaw, this took a long time and was super frustrating!) and was brought to a laser cutter. The board layout as shown by the outlines in Figure 1 was cut with the help of an Adobe Illustrator file. The board was then fit into the wooden frame.

This fitting presented another challenge. The board ended up too big, because wood compresses and warps over time, so we had to file down the acrylic until it fit snugly in the wooden frame (again, by hand). One problem with this is when I filed it down, I failed to recognize that the board was upside down, so that when Tyler put it in the housing and did the electronics, it turned out that the buttons were further from the player. More significantly, the last note (C) was now on the bottom, not the top of the board. That's why the pattern of notes discontinues for that high C. Once we got it filed and in, the only stuff left to do was secure the acrylic with a few screws and add some wooden blocks to get it to stay.

Additional construction of the board for the pedals involved hand-sawing and epoxying pedals, and sawing through the acrylic yet again to allow the introduction of the mode switching knife switch. The linear softpot was added by filing a hole for it through the wooden frame.

Wiring

Referencing Figure 2, the Arduino powered all the LEDs the same, without using digital out or analog out ports to control them individually. There was also not really time to get the overall change in brightness from the leds we would have hoped for, but it made sense to cut out non-musical functionality first. We would also have liked to selectively set the color or change whether the lights were on or off. Unfortunately, we would have needed more complex circuitry, as the digital outs were not powerful enough to control the leds as brightly as we would have liked. We ended up just using a set of equal brightness blue LEDS, made by snipping the extra leads of rgb connectors.

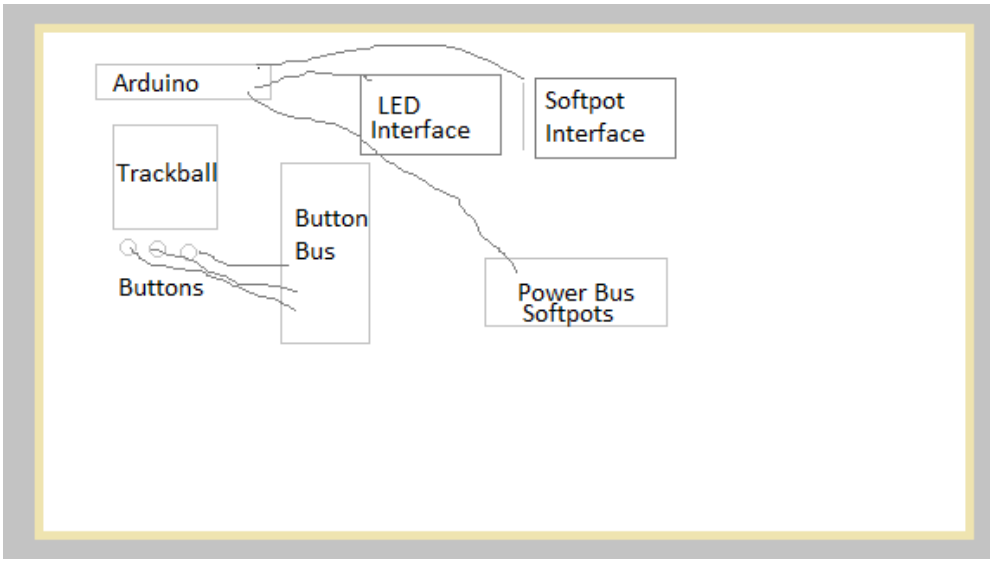

FIGURE 2: THE LAYOUT OF THE ELECTRONICS

The analog ports connected to the softpot interface, reading the analog position value of each softpot. The digital read connected to the button bus, which interfaced with each button and allowed their pressing to be detected. The sustain was measured in the same way, with a digital port reading whether it was on or off. The electrical diagram for the softpots, buttons, sustain and LEDs are shown in Figure 2.

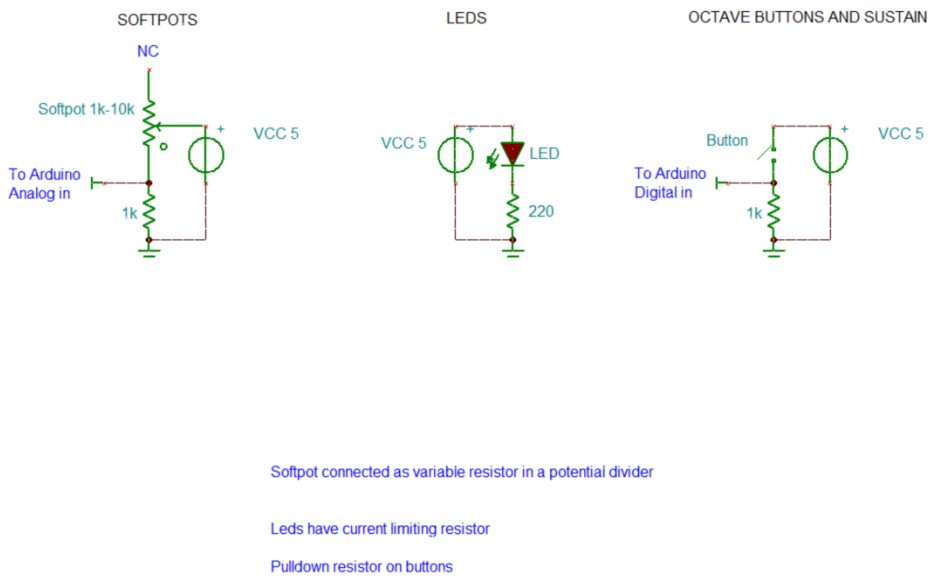

FIGURE 3: THE ELECTRICAL DIAGRAM FOR EACH COMPONENT.

One challenge we encountered in wiring was the softpots were so high voltage that you could destroy them by bridging the gap between the leads with your finger. This was solved by adding a potential divider that lowered the voltage enough so that if it did short it would still not blow out the device.

Programming

The Arduino

The Arduino to max patch was originally based off of the Arduino to trackball program that was provided online. This was merged with the example program that most people used to connect Arduinos, with some slight modification. The only other thing of note about the Arduino programming was that we ran into a problem where the trackball software didn't work with Windows, which was easily solved by switching to the Mac's in the lab.

Max

The max patch was split into 6 primary sections:

1. The main page with initialization and individual softpots velocity and note layouts configured
2. The trackball jit.phys module to do pitch shifts for the glass harp mode
3. The softpot patcher interpreting the stream of data from the softpots as position and velocity
4. Mode 1: Glass harp patcher
5. Mode 2: Theremin patcher
6. Mode 3: Sample mode patcher

1: Main page and initialization

In this section the data was read in and represented. It allows visual feedback on the data for each of the sensors, including the selected mode and which softpots are on and in which position.

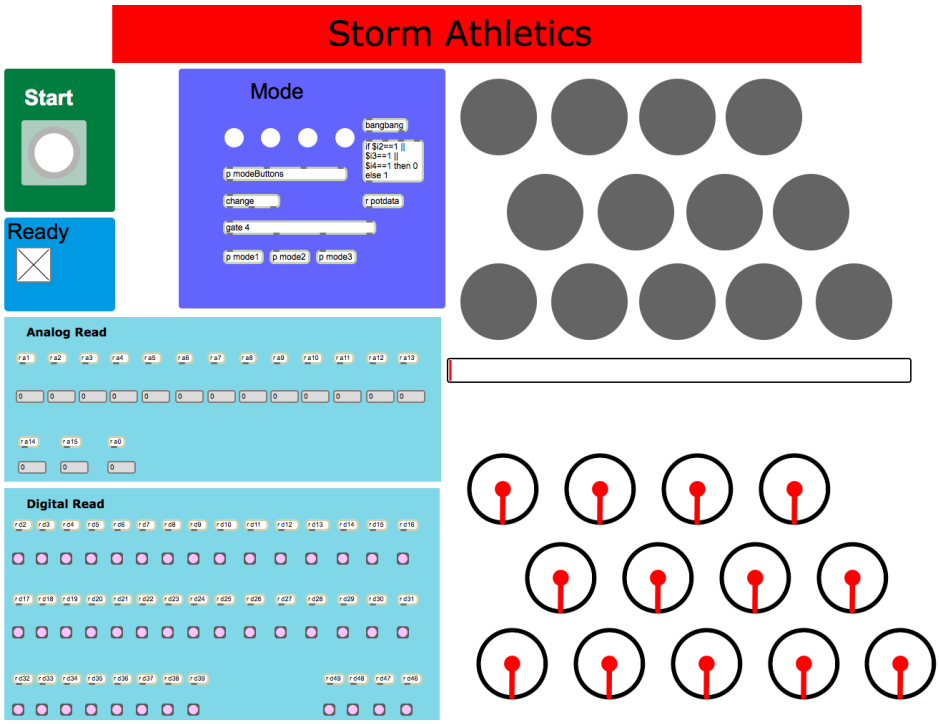

FIGURE 4: THE MAIN PAGE.

FIGURE 5: DATA IN FROM EACH SOFTPOT

As seen in figure 5, each note was read in and routed, sending the data through softpot patcher, potcontrol patcher which added the double-tap sustain feature, and out to the modes

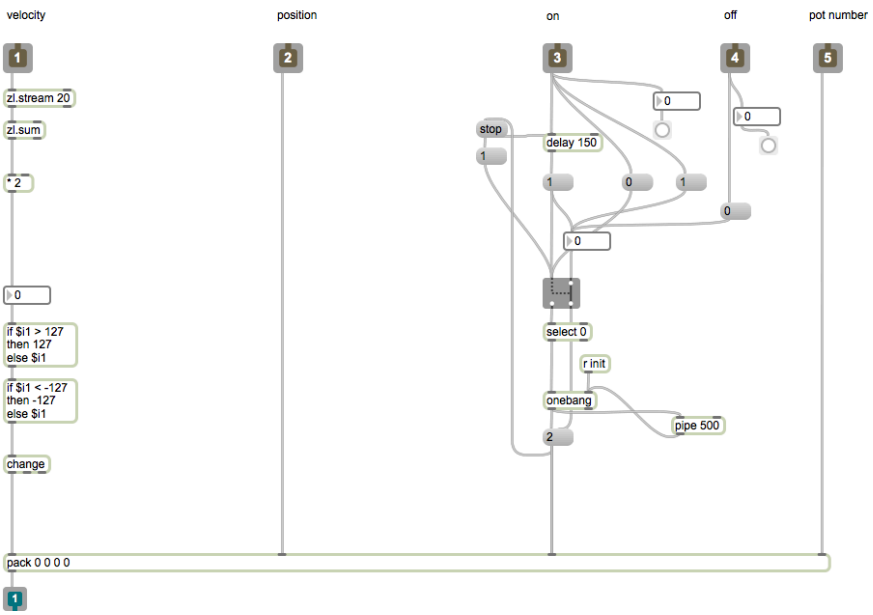

FIGURE 6: THE POTCONTROL PATCHER AS SHOWN IN FIGURE 5

2: The trackball jit.phys module to do pitch shifts for the glass harp mode

The trackball was built so that it would approximate the sound of a wine glass that was just accelerated and started to wobble back and forth in the glass. We wanted this effect to be as realistic as possible. This behavior was approximated with a ball rolling around in a wine glass with high friction, representing the center of mass of the water.

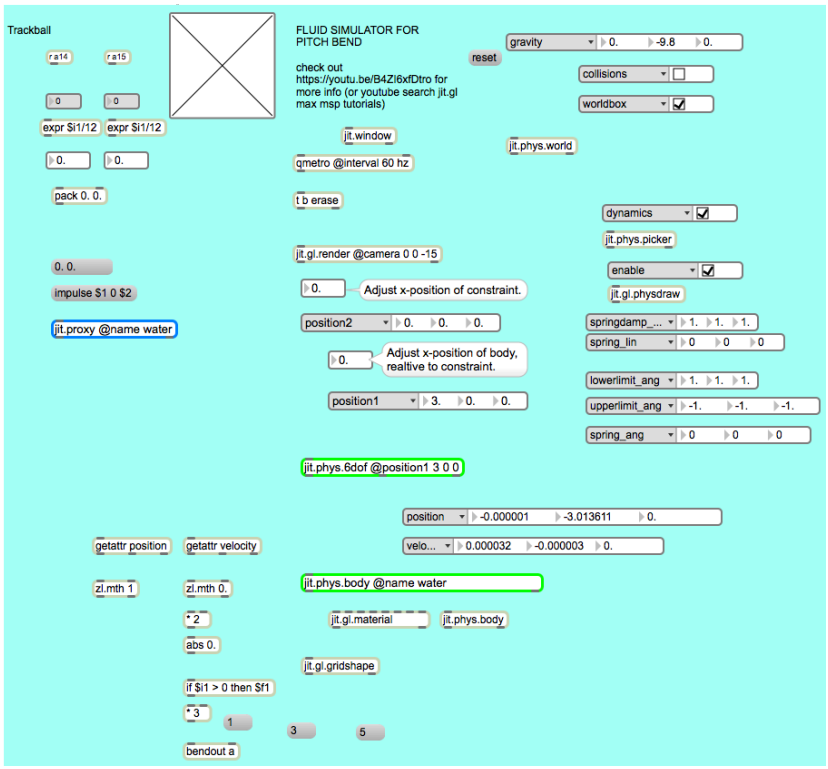

FIGURE 7: TRACKBALL DATA MAX MODULE

This effect was reproduced in max using Jitter physics, also known as the Jit.Phys object. The result was also visualized in a GUI produced by the program (figure 8). The input to the Jit.phys object was the velocity readings of the trackball, conveniently calculated onboard the Arduino using the trackball package mentioned previously. This velocity can be conceptualized as the velocity of a table moving the harp. When a cup of water moves, in the frame of the cup the side of the glass produces a force corresponding to the acceleration of the cup. This force can be approximated as hitting the center of mass of the water. Though we didn't use the acceleration of the trackball, the effect we programmed worked similarly.

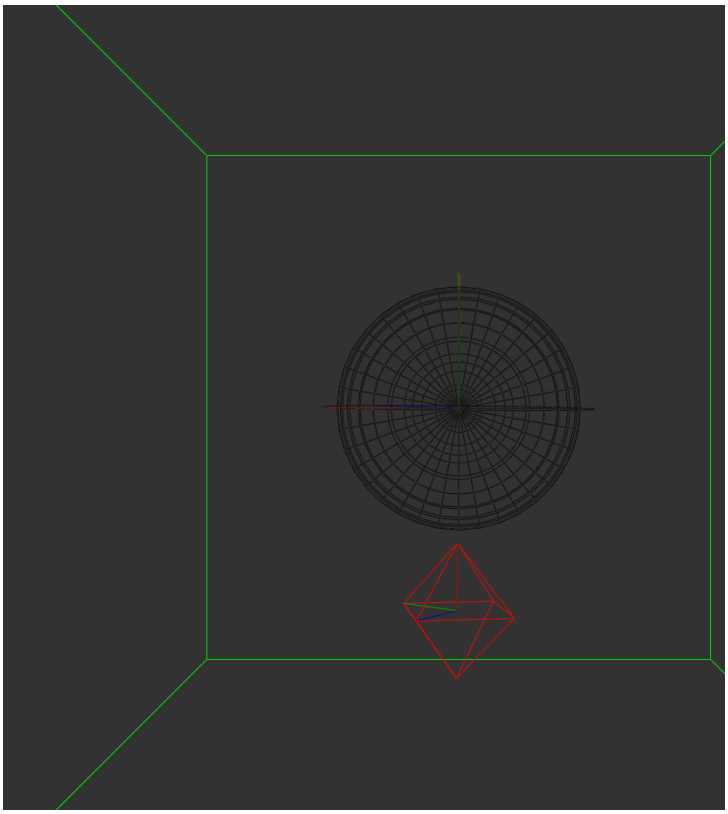

FIGURE 8: JITTER PHYSICS SIMULATION GUI OF WATER IN A WINE GLASS

The gravity in the physics sim was set to a very high value, and the default friction was used, as we ran into the problem that there's no way to change default air friction with jitter. A spherical constraint was used to get the ball to act like it was rolling around in a half-sphere shaped glass. The resulting motion is a damped 2d harmonic oscillation with a variable drive parameter in both x and y. The z component of the object is read in, and this is sent to a channel-wide pitch bend which affects all notes that are playing. Interestingly, unlike most pitch bends, the only way the pitch is bent is lower in pitch; the highest pitch the note we get is when there is no oscillation.

3: The softpot patcher interpreting the stream of data from the softpots as position and velocity

The softpots were definitely the largest programming challenge for our max patch. We encountered a couple challenges with the softpots. One of them was getting the data in to be linear with distance. As you get further along a softpot, the difference between position readings gets further and further apart (see the distances between the lines in figure 5). To correct for this, a fit determined by matlab was figured out to get each of the softpots reading equal distances per distance across the pot.

We also had to deal with what happens when you put your finger on and take it off the softpot. This was fixed by setting any velocity measurement to zero if your finger left the softpot, and the velocity determined by this jump was ignored by max.

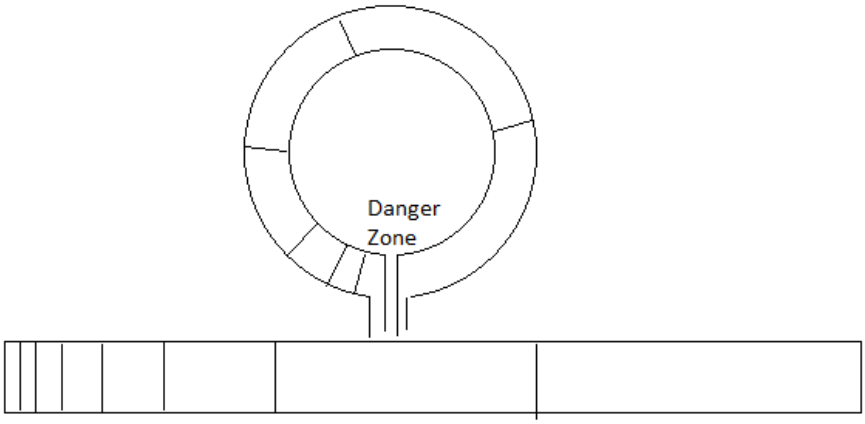

FIGURE 9: A SOFTPOT, AND WHAT IT WOULD LOOK LIKE IF FLATTENED OUT. EACH LINE IS EVEN IN POSITION READING AWAY FROM EACH OTHER.

Another challenge was the inaccuracy of the sensor as it read position values. There is a certain random jitter back and forth when you hold your finger on a softpot. We fixed this by adding a bunch of the velocity measurements (that included direction) together, so that small back and forth movements at high velocities would average out to zero. The sum also served to make the velocity reading when you took your finger off the soft pot lower gradually, as the control volume ignores release time in Reason. The max code is shown in figure 10.

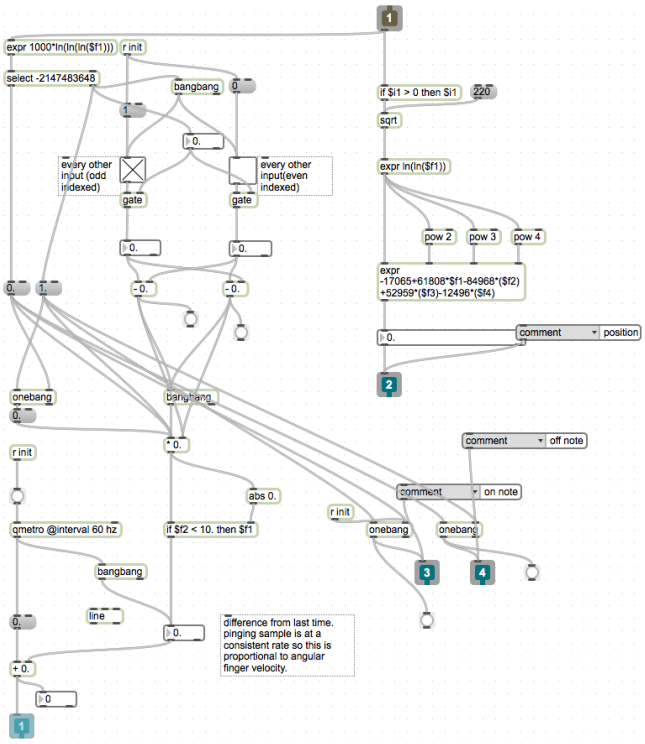

FIGURE 10: MAX CODE TO INTERPRET SOFTPOT DATA – SOFTPOT PATCHER

Finally, one of the the first challenges we faced was the bridge that connected one end of the component to the other, the “Danger Zone”. There is no way for the softpot to read position values at this point. Surprisingly, this was one of the easiest problems to fix. At the end of the velocity processing we applied a filter that cut off any velocity that was higher than a person could play. This effectively cut off any problems with bridging, because your finger is wide enough to never be completely off the softpot.

Once the velocity readings were calculated, we had to send it to reason to somehow change the loudness of the patch. The problem with this is we wanted polyphonic aftertouch, with the velocity of each softpot dynamically changing the volume of each pitch. We ran into huge problems with this, and spent many, many hours trying to get max to send fine-tuned loudness readings to reason. We eventually gave up and decided to simply get the maximum velocity and use that to change the loudness of the reason patch with a midi control signal for the master level of the patch. Later professor Lehrman informed us that the way to fix this is to give each signal a different label in Max, so that reason can parse out the midi control signals to the correct channels. This would definitely be something we would implement in the next version.

4: Mode 1: Glass harp patcher

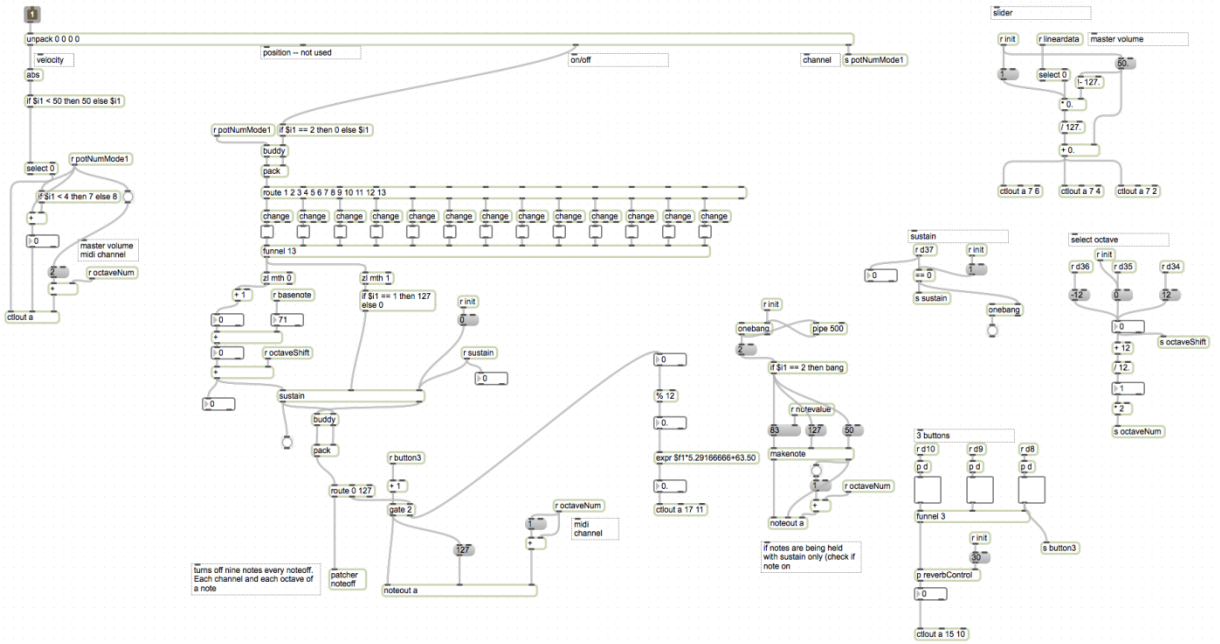

FIGURE 11: MODE1, THE GLASS HARP PATCHER

The glass harp mode works by unpacking an object sent through to it from previous parts of the program. The octave buttons used a simple message that sent -12, 0, or 12 to the pitches being played. It was programmed so that you could hold down a note and hit the button, and it wouldn't change to an octave up unless you played the same note again. This helped create the feeling that when you play the instrument, you have more keys than are physically present. The buttons modified a default value and added or decreased to a max or a min with enough button presses. Because of the three sets of combinators for this mode to achieve polyphonic volume control, there had to be three sets of control outs for every message to reason.

5: Mode 2: Theremin patcher

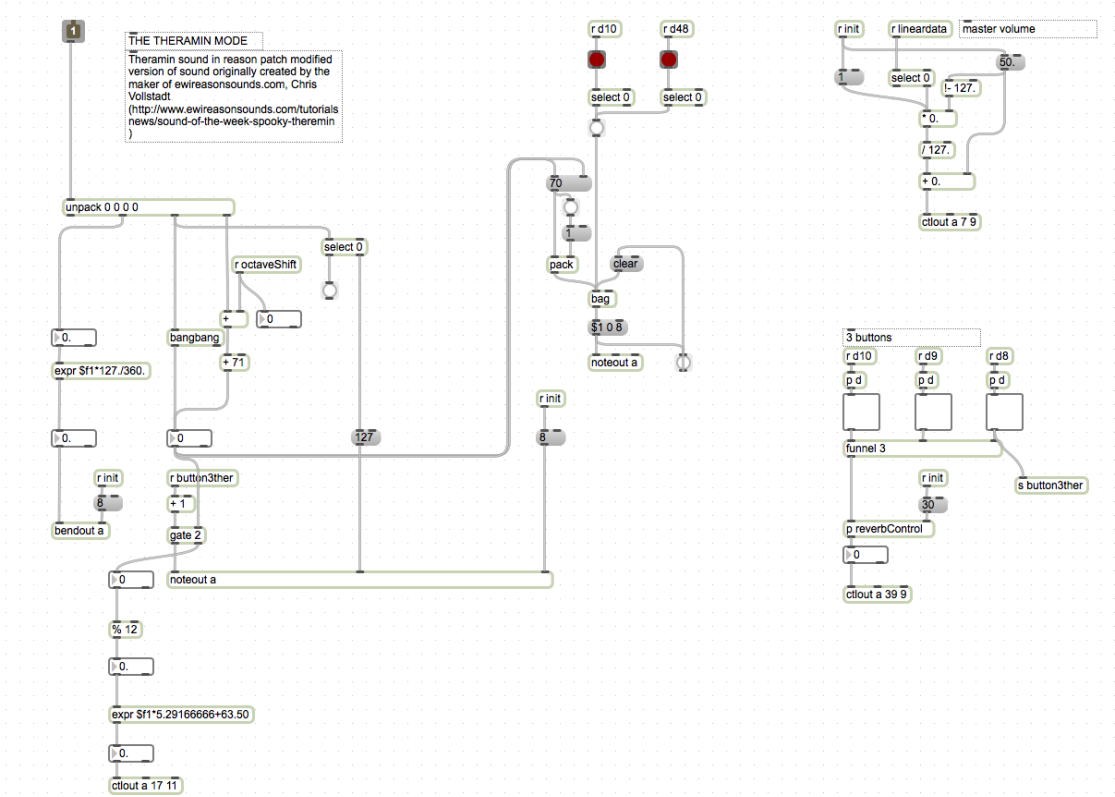

FIGURE 12: MODE 2, THEREMIN PATCHER

This essentially operated as a simplified glass harp mode without the need for sustain. It was the simplest of the three modes.

6: Mode 6: Sampler mode

The sampler mode contained complex calls to max and msp. The code can be seen below (Figure 13).

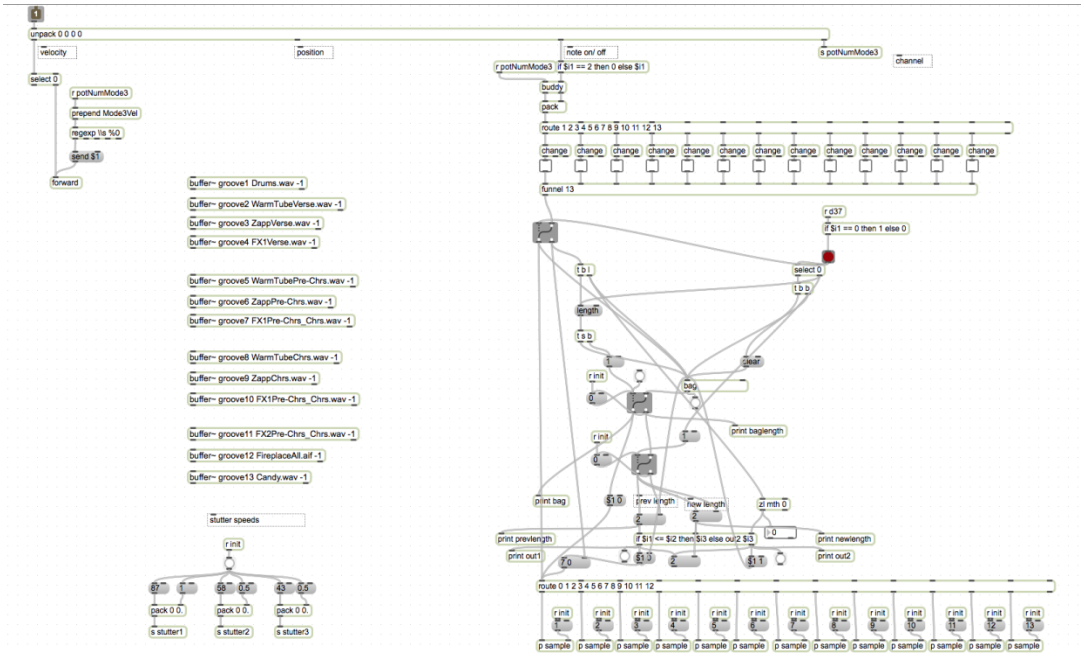

FIGURE 13: MODE 3, THE SAMPLER

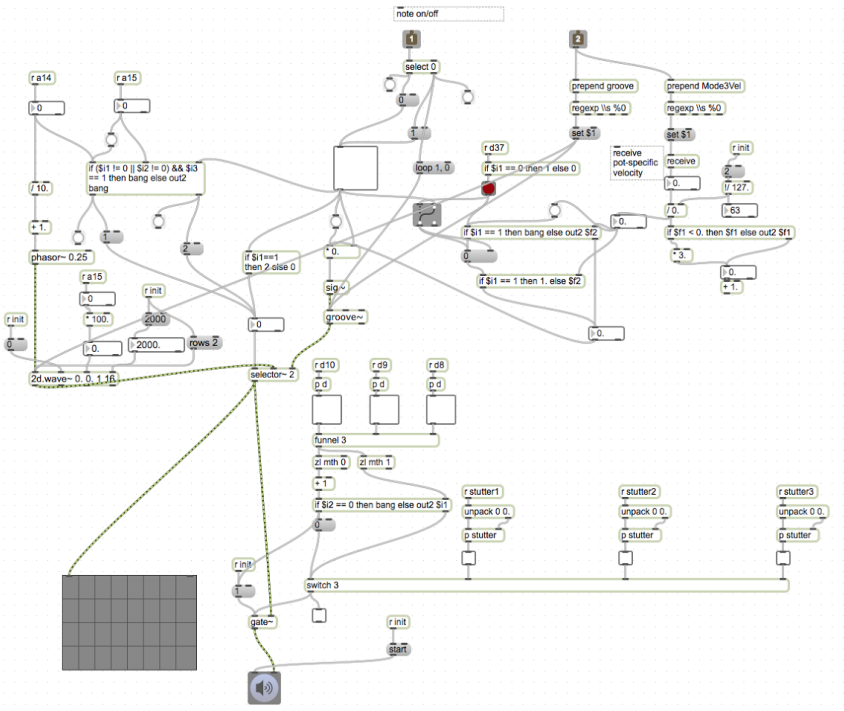

FIGURE 14: MODE 3, THE SAMPLER. INCLUDES NOTE ON AND OFF CALLS

Reason

One part of the reason patch was creating 13 subtractors so we could get each note on its own channel. We did this with one combinator for each octave, as is shown in figure 15.

FIGURE 15: COMBINATOR FOR EACH OF 3 OCTAVES

Getting a glass harp sound

The sound of a glass harp can be analyzed by using the slip and stick model: your finger sticks and slips away with each oscillating, setting up resonance in the glass. Wine glasses have the property that they have two resonant frequencies, which are very close to each other. In particular, looking at the thesis of Camilla Shu Yu Yang, (http://www.forskningsradet.no/servlet/Satellite?blobcol=urldata&blobheader=application%2Fpdf&blobheadername1=Content-Disposition%3A&blobheadervalue1=+attachment%3B+filename%3DYangCamillaShuYu.pdf&blobkey=id&blobtable=MungoBlobs&blobwhere=1274505957349&ssbinary=true) the fourier transform of the frequencies of standing waves in the glass show a difference in frequencies of about 3 hz when at a high pitch (~750 hz). This causes a beating frequency of 3 hz, which is what you hear when real glasses play.

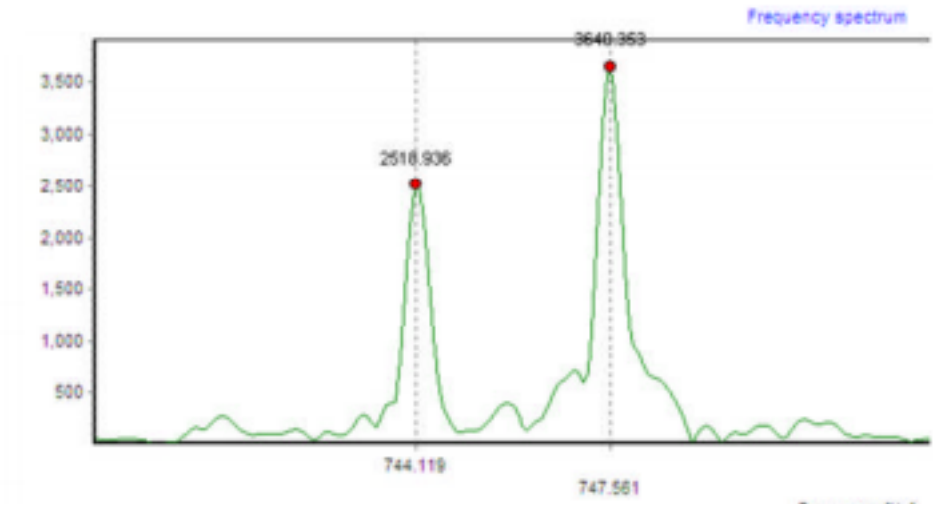

FIGURE 17: THE FOURIER TRANSFORM OF THE FREQUENCY SPECTRUM OF A GLASS HARP. INTENSITY IS THE Y AXIS.

Converting this to difference in pitch, this is about 7 cents (hundredths of a semitone). A few more physical effects for a glass harp can be found from this analysis. One of them is the application of a bandpass filter just to make sure not too many overtones are produced. Another is that the exact frequencies are variable over time and go in and out of phase, as any messy physical system would. This takes an edge off the beat frequencies and makes them less noticeable than if glass harps were perfect oscillators.

We approximated this effect with a single reason patch with two sine wave oscillators (Figure 17). Each was set to 3 Hz apart, more appropriate for a lower absolute pitch, as the difference between the peaks in frequency get further apart with lower absolute frequencies.

FIGURE 18: SUBTRACTOR FOR THE REASON GLASS HARP SOUND

The other changes were a phase and FM change, along with a mix. This made it sound more “glassy” and got rid of some of the annoying precise beating that the simplistic application of the two pitches with no

modification will produce. Not much else was used to mess with the reason patch, as it turns out the acoustic signature of a glass harp is both simple, and with enough listening, immediately recognizable.

Conclusion

We see with electronic instruments like the Marimba Lumina that people who get to a certain level of proficiency at an instrument want to take it further. Professional glass harp players, who already have extreme skill with their instruments, now have a whole world of possibility for performance and expression they didn't have before. None of us have actually played before but I think it really approximates well the sounds and feel of a real glass harp. Not only that, the flexibility of the instrument and many more possibilities for its use made all of us really excited to get it built and play it!

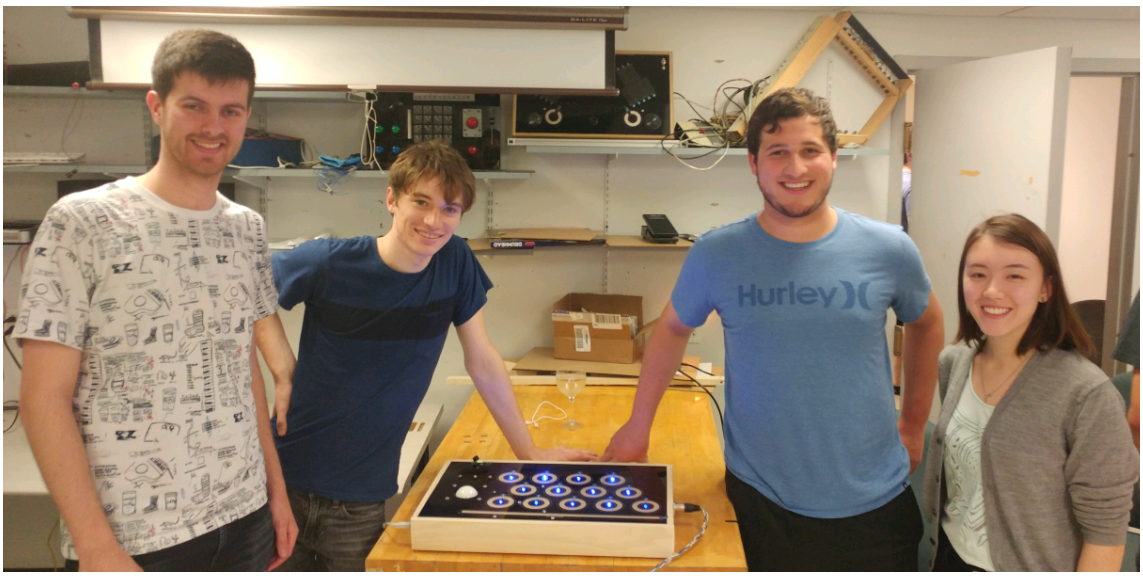

FIGURE 19: JACKSON, MORGAN, TYLER, RACHEL, AND GLASS HARP MIDI CONTROLLER