

Trumpet Wind Controller

Matthew Kelly

Justin Griffin

Michael Droesch

Design Proposal / Concepts:

The design proposal for this project was to build a “wind controller” trumpet. The performer controls the volume of the note by blowing into a MIDI pressure sensor. Three touch sensors act like the valves in a traditional trumpet, and the harmonic is selected by a rotary potentiometer. Trumpet players often are required to play with a mute or switch to flugel horn; this MIDI trumpet allows the performer to switch between several voices. Another effect which is commonly requested of trumpet players is to “play into the stand” or in some other way effect the timbre of their instrument by moving it. A 1g accelerometer is built into the trumpet to allow the acceleration (including orientation) to be mapped to a MIDI parameter. A linear potentiometer is used to control vibrato.

Construction:

The body of the trumpet is made up of a short section of a standard “2x4”. Three 1¼” diameter holes were drilled in the top so that the buttons could be inserted. Small holes were then drilled into the sides of the large holes to allow for wires to be connected to the buttons. These buttons acted like the valves on a standard trumpet.

Two smaller buttons were mounted on the side of the main 2x4 so that they could be easily pressed by the performer’s pinky finger. These buttons cycle through the list of instrument voices which can be produced by the controller. The three main buttons as well as these two selector buttons can be seen below in Figure 1.



Figure 1- Right side of the trumpet controller. The accelerometer is located beneath the blue foam sheet.

A large hole was drilled in the wide side of the 2x4. A rotary potentiometer was mounted in this hole, and a second hole was drilled through the bottom to route the wires through. A small binder clip was fitted to the shaft of the rotary potentiometer; this allowed the relative position of the shaft to be adjusted if necessary. A small piece of wood was cut such that the outside was a circle, and the inside had a cutout which was just the right size to fit over the binder clip. This allowed the performer to be able to easily control the rotation of the potentiometer. Small nails were driven into both the body of the trumpet and the small piece of wood to restrict the rotation of the potentiometer to a comfortable region of roughly 120 degrees. Figure 2 shows the rotary potentiometer side of the trumpet.



Figure 2- Left side of trumpet controller

The accelerometer was mounted to opposite to the rotary potentiometer. A small circuit (salvaged from an old project) was used to provide the correct excitation voltage to the accelerometer and do some hardware noise filtering. The accelerometer is beneath the blue foam as seen in Figure 1.

A second 2x4 was cut in a unique shape to enable the linear potentiometer and breathe controller to be mounted to the instrument. This second piece of wood was cut so that it formed an angle with the main structure. The bottom of this piece was curved so that the performers thumb can easily fit under it. The linear potentiometer was mounted to the top of this new piece of wood, and holes were drilled to allow it to lie flush with the surface and route the wires through the structure. The Yamaha breath controller was mounted to the far end of the second 2x4 in a location similar to where the mouthpiece would be on a standard trumpet. Figure 3 shows how the instrument is held by a performer.



Figure 3- Trumpet Wind Controller being played

Max Programming:

On an acoustic trumpet each valve has a short length of pipe attached to it. By depressing each valve the corresponding length of tubing is added to the length of the instrument, dropping the pitch. Table 1 gives the valve combinations and corresponding note change associated with it. The number of half steps is the number below the note being played in the open position. Depressed valves are noted with “X” and open valves are left blank. The note code is used in signal processing and will be discussed later.

Table 1- Valve combinations

# Half Steps	0	1	2	3	3	4	5	6
Valve 1			X	X			X	X
Valve 2		X		X		X		X
Valve 3					X	X	X	X
Code Num:	0	10	1	11	100	110	101	111

Each button which is acting like a valve is read into max as either 0 or 127. The max patch (shown below in Figure 4) reads in each button, and assigns each button event a number as shown in the list. This patch has the net effect of sending a bang message whenever a button is pressed or released.

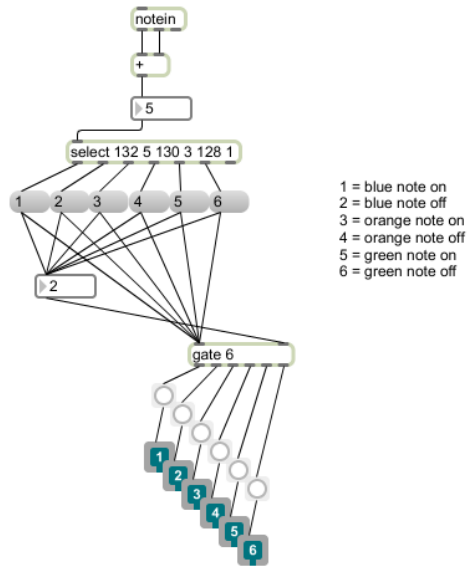


Figure 4- Max Patch (Note Find) to control valve (button) signals

The bang messages for each button are run into a toggle switch so that the state of the switch perfectly corresponds to the state of the buttons on the trumpet. When a button is depressed the toggle switch sends a bang message of 1, and when it is released it sends a bang message of 0. For button two this value is multiplied by 10, and for button three this values is multiplied by 100. A select function easily allows these code numbers to be converted to the number of half steps to reduce the note by. See Appendix 1 for the main max patch.

On an acoustic trumpet the performer can change the harmonic that the instrument is playing by adjusting their embouchure. This is very difficult to measure so an alternative method has been devised using a rotary potentiometer. Most of the playing done on a trumpet is done in the first 6 harmonics (excluding pedal tones). The output of a rotary potentiometer can be scaled so that it cycles through the values 0, 1, 2, 3, 4, 5. Excluding the pedal tones, the open harmonics on a trumpet are: Bb, F, Bb, D, F, (A*), Bb. The A* harmonic does not easily fit into the European scale and is typically avoided, and has thus been left off the list used in this program. Each harmonic number was scaled to correspond to the number of half steps above the fundamental.

The rotary potentiometer used in this project produced a highly nonlinear response which made it very difficult to play when the midi notes were simply scaled. The MIDI output versus angle plot was made to determine the response curve for the potentiometer (see figure 5).

Angle	Midi
0	0
10	0
20	0
31	10
45	27
60	49
79	73
92	92
106	100
121	107
141	112
155	115
178	117
194	120
211	122
228	123
247	125
262	126
285	126
304	127
318	127

Angle	Midi Num	Harmonic
31	10	
		0
51.7	36.8	
		1
72.3	64.5	
		2
93	92.6	
		3
113.7	103.6	
		4
134.3	110.3	
		5
155	115	

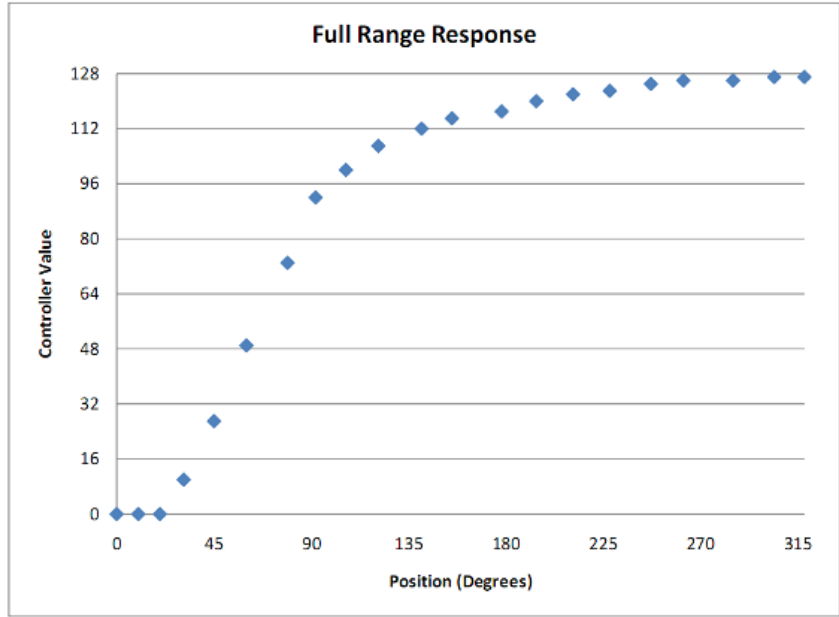


Figure 5- Rotary Potentiometer response Curve

In the setup used for this project it was comfortable for the performer to rotate the potentiometer by about 120 degrees. The potentiometer is unresponsive below about 30 degrees, so that was set as the minimum angle. The 120 degrees after this were broken up into 6 equal segments, and the midi numbers corresponding to those angles were determined through linear interpolation. A series of logic gates were used in max to convert the MIDI number output by the potentiometer into a linear response for the harmonic number. The end result for the performer is that each harmonic change occurs after about 20 degrees of rotation so that the instrument is predictable. The physical construction of the instrument prevents the rotary potentiometer from rotating below 30 degrees or above 150 degrees.

A small max patch (see Figure 6) was written to combine the output of the harmonic selector and the valve selector patches. The low C (Bb concert) on a trumpet is being used as the fundamental here and has a MIDI number of 58. This number is added to the harmonic shift number, and then the number of steps produced by the valve combination is subtracted. The result is a single MIDI note number which is output to Reason.

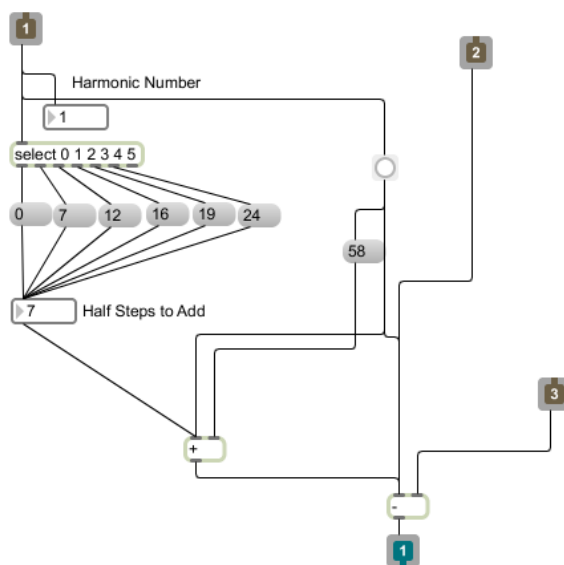


Figure 6- Patcher Harmonic- Max patch which is used to combine the outputs of the buttons and rotary potentiometer to produce a note

The linear potentiometer data is sent directly to reason to manipulate the LFO. The outputs for two of the three axes of the accelerometer are sent to a visualizer program which works similarly to an “etch-a-sketch”. The third axis accelerometer data is sent to Reason to control the chorus flanger.

The breath controller is read into max, after some minor adjusting, to control the main volume level. There are several adjustments which can be made on the hardware by the performer which make additional manipulation in max unnecessary. When the gauge pressure increases above zero a makenote command is send to Reason. As the gauge pressure level again drops to zero a flush command is sent which turns off all notes. This allows the performer to articulate notes as if playing a real trumpet. As a note is being held it is easy for the performer to increase or decrease pressure to adjust the notes volume. There is also a section of code which creates a new note whenever the button combination is changed, allowing the performer to play legato by changing fingerings.

The max patch can output on any one of four channels which are controlled by the two small buttons near the performers right pinky finger. The buttons simply increment or decrement a counter which then outputs a channel change. A flush command is sent whenever the channel changes to prevent notes from hanging over.

Reason Patch:

The reason patch for this project was fairly simple. An NN-19 digital sampler is the basis for each channel. The output of each of these modules is then fed through a RV-7 reverb module and a chorus flanger. Everything is then routed through a line mixer so that the master volume can be controlled. The LFO for each module is adjusted with the linear potentiometer, and the accelerometer is used to adjust values in the chorus/flanger. See Appendix 2 for a section of the Reason Patch.

Final Performance:

As a rough instrument the trumpet worked fairly well. I was able to play a melody (Theme from the largo movement of Dvořák's New World Symphony) successfully. After some practice it was possible to attain a fairly high degree of control over dynamics, as well as slow legato for notes in the same partial. The wind controller trumpet was also able to successfully play faster pieces which were staccato, and was able to readily switch voices.

There were several problems with the performance. The first problem is that, given current programming, the trumpet is unable to play legato across a harmonic change. The air stream must be completely stopped in order for the harmonic to be changed. A less obvious problem is that the computer takes an appreciable amount of time to switch notes when the buttons are pressed. This delay isn't really perceptible until a quick run is attempted, at which point the notes lag. This lag can be overcome by rearticulating every note, but this effectively prevents legato runs from being performed.

The second major problem is that the LFO and Flanger controls did not function as per the original design. When the LFO was turned up it would get fed back through the reverb creating an interesting, but undesirable noise. The flanger control (corresponding to the accelerometer) worked, but the effect was almost unnoticeable.

Future Work:

If I had more time to work on this project I would fix the Reason patches. As a trumpet controller it would make sense to have trumpet, flugel horn, and muted trumpet available as patches. In addition I would include a violin and vibraphone to get some different sounds. By removing the reverb patches it would be possible to quickly fix the LFO control. The accelerometer could be routed to a different control for each voice, allowing for a variety of effects. It is important that the performer be able to play a steady tone, so I would modify the reason patch to scale the accelerometer input. Figure 7 shows a possible way to do this. All of the midrange values are mapped to a single MIDI number, and when the input leaves the midrange it scales more quickly. This prevents unwanted effects, but allows them to work well when desired. For some voices it might make sense to map this control to pitch bend; I think that this would work well for the violin voice. Another possible addition would be to use the accelerometer to control a filter frequency or a reverb.

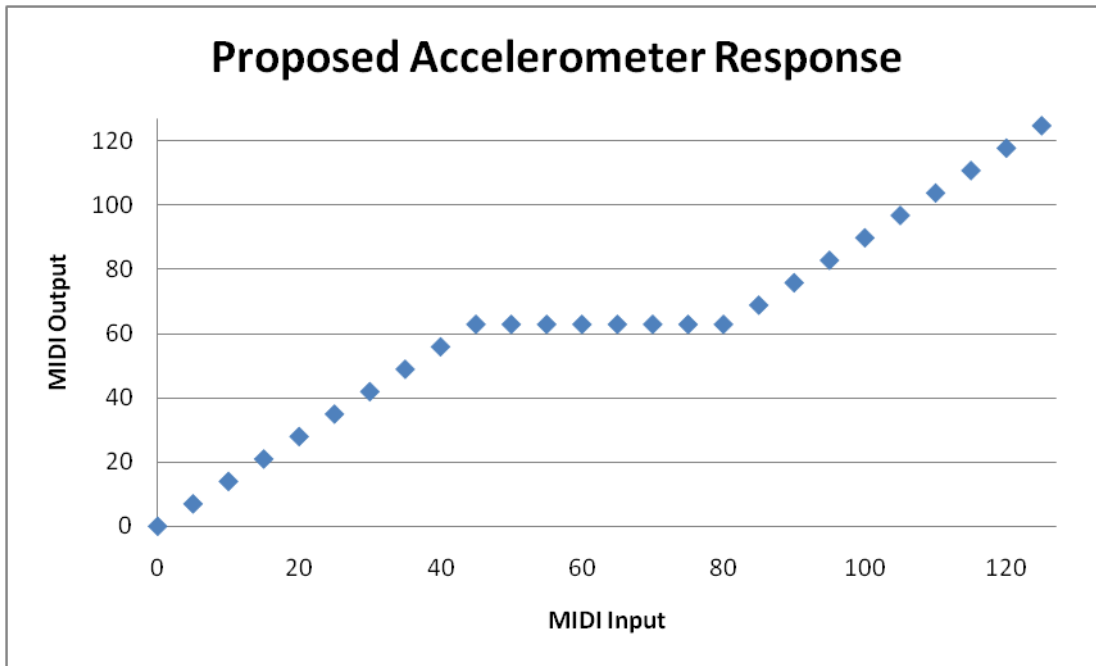
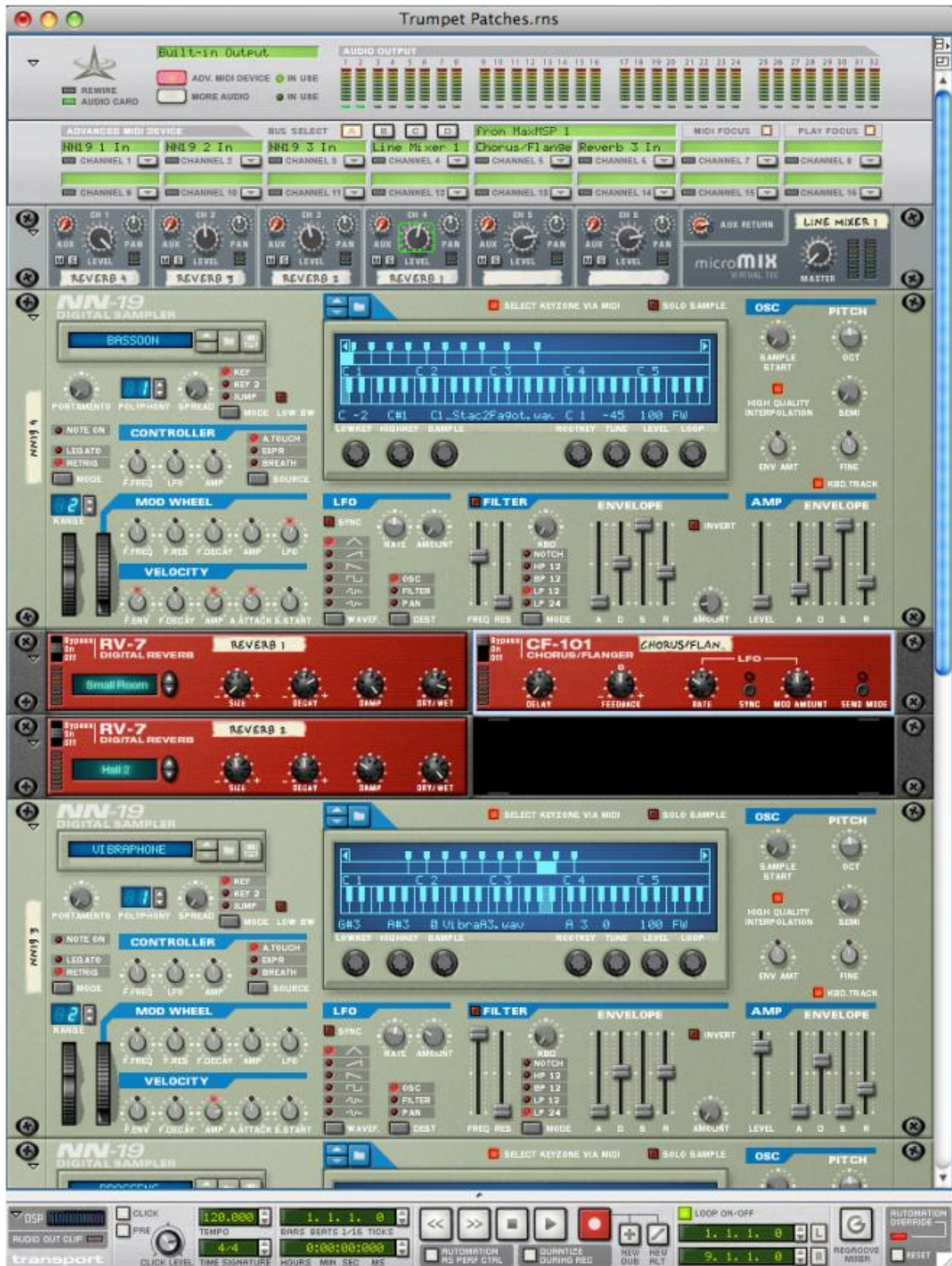


Figure 7- Proposed Accelerometer Response

It would be possible to adjust the max patch to allow for legato across a harmonic change. A pipe could be used to compare previous harmonic values to the current one. When a change is detected a bang could be sent to rearticulate the note. This could then be run through a 1-bang function which was set to 19ms. This corresponds to a 16th note at 200 beats per minute which is about as fast as anyone would want to play.

For the most part the instrument was easy to play but two minor adjustments would significantly improve it. It is very easy to overshoot the desired harmonic as each harmonic corresponds to a range of 20 degrees. If it was possible to have the wheel “click” as it changed harmonic that would enable the performer to play the instrument without having to continually watch the display. The simple way to achieve this would be to have a series of notches which a small piece of plastic ran across at each switch. This would be relatively inaccurate, and a “click” may not perfectly align with a note change. A second solution is to buy a 6 position switch. A final, more complicated solution would be to use controls to produce the desired effect. It is possible to build a rotary switch which provided haptic feedback for any number of desired positions for a shaft connected to a motor. The computer simply makes the position of the wheel stable at 6 positions, forcing it to snap between them. Since the number of positions is controlled digitally, the user would be able to change the number of harmonics and the angle between them. While this would be a fascinating solution it would likely be cost prohibitive and nearly impossible to implement in max. A second, far simple change I would make to the trumpet is to include a wrist strap to help hold up the instrument. This would in turn make the vibrato control far simpler to use.



Appendix 2- Reason Patch

Team Assessment:

Matthew Kelly: I was responsible primarily for the construction of the trumpet. I designed the frame of the trumpet, how it would be played, and how to attach all of the sensors so that they were easily accessible. Additionally, I was the one with the most music theory and brass instrument experience so I also was put in charge of determining note values. I wrote the max patches that convert a combination of three buttons into a chromatic decrement number, and the sections of the patch dealing with the harmonic number. As part of the harmonic number selector algorithm I determined the response curve for the rotary potentiometer and determined how to linearize it and get the best possible response. I also had the responsibility of learning to play the instrument for the final performance

Justin Griffin: Justin was primarily responsible for writing the max patch. He combined the patches that I wrote with the rest of the program. He wrote the sensor in / out code for all of the sensors excluding the three main buttons and rotary potentiometer. He also wrote the code to determine note articulation, and channel selection. The visualizer which was used was taken from a previous project of his, and he designed the front panel. We did a lot of work together debugging both the hardware and the software to get a playable instrument.

Michael Droesch: Mike was responsible for working on the Reason patch. Very early on he did a little work on the hardware, mainly determining which accelerometers and linear potentiometers worked. He did not put as much time or effort into the project as Justin and Myself. That would have been ok if the Reason patch worked well, but it was very buggy. The LFO did not work as it was supposed to, and the accelerometer barely had an effect on the sound produced.

Group: Justin worked very hard and was great to work with. He put a lot of effort throughout the project especially near the end fixing bugs in the program. While Mike did some work, and produced an acceptable Reason patch he put in almost no time towards the end of the project. The problems with the Reason patch were apparent a week before the performance and Mike put in no effort to fix them even though it was his responsibility.