Samuel Paré-Chouinard
CART 360 Final Project Prototype
https://github.com/miwamiwa/CART-360/ThisThenThat_Prototype

Folder contents :

- old code (folder) : previous iterations of the max and arduino files. Mostly for my own reference.
- Documentation.pdf : this pdf. Contains answers to the questions asked and documentation of the prototyping process.
- Bigschematic.jpg : schematic of the entire system
- maxtalk4.ino : arduino code
- maxtalkin3.maxpat : max patch to run
- NewPing_v1.9.1.zip : library required for arduino code
- Smallschematic.jpg : schematic of the circuit only

PART I - Questions

A Non-Technical Written Response which addresses and relates the implicit concerns of Why Do We Prototype? & Fidelity Levels to the development process of your Physical Prototype. (MINIMUM 512 WORDS)

In the exerpts from Kathryn McElroy's book there is talk about prototyping, and prototyping alot. McElroy's advice is to use prototypes to test out ideas and directions a project can take. She talks about different scales of prototypes that can be made, discovering new ideas in the process and discovering which ones don't work and should be dropped. She stresses the importance of using the prototype to envision how the end user will experience your work, and the importance of iterating until the best possible solution to a problem is found.

Though I must admit to pushing off reading the exerpts until the end of my prototyping process, I find there are similarities between what McElroy describes and what I did. I count 8 different experiments using my breadboard and motors, and eventually max msp and an ultrasonic sensor. Though they weren't all very complex, these could all be considered individual prototypes as everytime I rebuilt a system entirely, using knowledge gathered from the previous prototype to add, change and remove different parts. Each prototype was a test for how a system would perform or sound, and in effect each iteration re-shaped the direction of the project in its own right.

Most of these prototypes are quite low-fidelity. Even the prototype I am turning is a bit hard to describe as being "medium-fidelity" since it doesn't include many features for the user to interact with yet. It includes a preliminary exploration of what that could be - some kind of use of one (or many) ultrasonic sensor(s) - but it doesn't seem to answer many questions related to interactivity just yet, on the surface at least. Behind the decor, there are actually quite a few performance-related questions that I've managed to clear out. By iterating on my communication protocol between arduino and max/msp, I was able to build a code that can deal with sensor inputs and motor motion in real time, without any interruptions. It's actually a non-negligible aspect of the design, since I now have doors wide open to create a dynamic music making system driven by sensory inputs, rather than being limited by waiting for sensory inputs.

I set out on the idea of producing some kind of interesting sound source using mechanical motion. My initial thought was to layer rapidly tapping hammers to create a unique sound texture, which Sabine responded to by suggesting that I try using max/msp in combination with arduino. With these two vague ideas in mind I built my first two prototypes: basic hammer motion using a servo, and communication between max/msp and arduino. From that point I was able to blend the first two prototypes into one and do some initial experiments with how to control things in max msp. From there I was able to carry through to my original plan, and scale that prototype into a fourth one which featured three servo-driven hammers controlled using a max patch.

A Technical Evaluation of Sensors and their associated Affordances which would ideally support your project's proposed Interaction Design Strategy (MINIMUM 512 WORDS).

There aren't very many sensors in my design so far, but I do intend to add more. I will answer the question by describing the parts in this prototype and talking about what kinds of other sensors I could see being a part of this project.
There are five main components to my prototype: the servo motors, the ultrasonic sensor, the arduino (and its serial), max/msp, and the mic and speakers (or headphones).
The motors are the main source of sound in this project. I found servo motors interesting because they allowed me to design and tweak a specific motion. Physicality and regularity are the main reasons why I chose motors to produce sound. In effect they create a direct link between my project and the place in which it is located. Tapping objects seemed to me like the most effective manner of making sounds in the environment which initially inspired this project (mostly consisting of concrete and metal).
The ultrasonic sensor should allow me to create a numerical input to max/msp that is space-driven and motion-driven. The interaction which I want to create is one that would be reminiscent of playing with a set of water jets, so I want sensors that can pick up on people's presence with the space, and perhaps people's motion in it too. The ultrasonic sensor seems like one way to do that.
The interaction between arduino and max/msp allows me to create a dynamic work that is shaped in real time by a user's inputs via the sensors, and orchestrated in a complex way. Max/msp allows me to refine the sound which is being produced by the servo motors by allowing me to trigger different speeds in sequence, creating slow rhythmic patterns just as well as layers of rapid tapping. The mic and speakers/headphones allow to me spin the rudimentary sound I produce into something more affecting. As it is now I use some sound treatment to make the tapping sounds appear more similar to splashing sounds, which should hopefully work to support the theme further.

In terms of sensors that could be added to this project, I see a few:
Firstly I could add transreceivers to give the project some modularity. Sensors could make up a system of their own, together with an atTiny and a transreceiver, and sensor systems would communicate back to the main arduino, which would relay messages to max. It wouldn't be incredibly hard to design seen as communication between arduino and sensors would be one-way.
With a light sensor placed where a water jet would be, I could attempt to reference the act of blocking a water jet at its source by introducing some kind of sound effect when the user places his hands over the sensor.
Judging from its technical description, the Pyroelectric infrared motion sensor sounds like it would be

best suited for the motion detection aspect of the project. It detects infrared signals coming from moving people. With a detection angle of 100 degrees and a detection distance of 7 meters, I can imagine that a few of these could be placed in a manner that covers alot of ground. Having their range overlap could also create an interesting scenario where more complex messages are sent into max msp. Without doing any kind of position calculation, combinations of triggers could be used to create more unexpected sounds. The ultrasonic sensor still appears interesting as it returns a (somewhat) precise measure of distance, while this circuit returns binary info it seems (motion or no motion). Using a combintion of both might be the way to go.

Has your Project's initial intention or supposed meaning changed over the course of researching and implementing the Physical Prototype? If YES or NO – Explain why? (MINIMUM 256 WORDS).

My project has definitely evolved to a certain extent while I was prototyping. By working with different methods of controlling my servos using max, I got to explore the limitations of the method of sound production which I had chosen. On the flip side, I also got to explore the range of possibilities of how I could use max/msp to augment my project. What I have in my prototype is an amalgame of raw sound taken directly from the space, and sound treatment intended to reference the element (water) which is usually used playfully in that space. The intention shifted from my original focus on the material sounds available in the space, to a new goal more oriented towards recalling the activity the space is used for. By introducing digital sound treatment, I can reference directly what is missing in the space - water. I think it also contributes to create a more playful experience than purely rhythmic sounds, play being one of the main themes I put forward in my project proposal.
My initial design for this project was rather vague so I could also argue that it hasn't changed much over the course of building the prototype, it has simply become a bit more refined after some experimentation. The introduction of the water sound is a definite shift from what I originaly intented though. I doubt the project will sound/act the same once I am done adding in sensors. The intention of creating a playful reference to water jets in a public pool remains a firm orientation for the project however.

PART II – Notes on the experiments/prototypes, and documentation.

Final prototype : https://youtu.be/5ceoe2wco_I

The following are notes I wrote to wrap up my thoughts on the project. There are videos for most of the steps in the process.

>>>>>>>>>>>>>>> proto 1 october 13:
https://youtu.be/bLmnhcubRQk

testing servo as a percussive tool: hammer moving down and back up. There is a button but note how I am pressing the reset button :D uses arduino only. moves rather slowly. Here I taped the servo to a couple of hockey pucks to weigh it down and allow for as much motion as possible. I end up keeping that configuration for a few iterations, up until I change the hammers.

>>>>>>>>>>>>>>> proto 2 october 13:
https://youtu.be/TjgmKtbt8jI

testing messaging between arduino and max msp, using a youtube tutorial.
communication works both ways (button press sends numbers to max, and clicks trigger an led on the breadboard), but it is asynchronous.

>>>>>>>>>>>>>>> proto 3 october 13:
https://youtu.be/dGC0VQqBEbs

testing the first two prototypes together. button press sends a trigger to max which returns two target positions for the hammer to reach, one after the other. some shy attempts at achieving a faster hammer motion.. No notable communication issues here, but the test is also very simple

>>>>>>>>>>>>>>> proto 4 october 13:
https://youtu.be/TA_vyiMiKDI
https://youtu.be/JhC7hEdv8z0

multi-servo motion controlled via max msp. this is really just the previous prototype times three. I even tried having three buttons to control the hammers. the goal of this one is to explore how max could be used to create an interesting noise texture out of the three servos. I patched a bit as I was experimenting, and tried tapping different objects as well, with different types of metallic and plastic sounds.

the max patch i built for this one was meant to make a kind of strumming effect by playing them out of phase. the effect is approximative and drowned out by the noisy hammer motion. Some level of speed is achieved, which is nice because I was looking to create a kind of tremolo effect.
As the session continues I shorten the hammer motion and get much more speedy hammer motions, with much less motor noise. Pushing even further in that direction, I start to get quick strumming on all hammers. Very satisfied with how fast the communication on the serial port seems to work.

First issues with two-way serial communication without handshaking happen here, but I'm not sure what to do about them yet. In retrospect it's clear that messages are just getting mixed up in the serial. I give up the idea of sending button inputs, not knowing what to think about it. Instead, I focus on one-way communication from max to arduino that will orchestrate the tapping.

>>>>>>>>>>>>>>> proto 5 october 18:
https://youtu.be/Hu9g8FXTc2I

there is one short video of prototype 5. it is essentially a rebuilt version of prototype 4 that i made to test out some improvements in my max patch and another possible method of controlling the hammers using the function objects. Didn't really like it in the end. This play on phase is starting to feel a bit lame too. I also added a second type of message from max to arduino which adjusts the hammer height, so that I can configure them freely before starting or during the testing.

>>>>>>>>>>>>>>> proto 6 october 27:
https://youtu.be/sMTK4p5J3GY

changed hammer from one-sided to two-sided. the servo hold it from the middle of the shaft now, and it is taped to a small pile of quarters, all together taped down onto the table. Faster tapping indeed. There is also a new max patch to control the hammers, which uses the multislider object allow for programming different hammer speeds in a loop. There is no more need for the height adjustment message i had just added. I like the new max patch, now it's time to add the ultrasonic sensor.

>>>>>>>>>>>>>>> proto 7 november 1-2:

tested the ultrasonic sensor on the side, now it's time to add it in. And... it doesn't work. Serial communication is going ham, hammers are twitching slightly rather than joyfully tapping. I do some research and first I find out about handshaking, so I implement that but it doesn't help. I take some time to find out how the pulseIn() function I was using to read the ultrasonic sensor delays the arduino for about half a second every time. I take some more time to find a library that can help me, and try again. There is no documentation for this prototype.

>>>>>>>>>>>>>>> proto 8 november 3:
https://youtu.be/5ceoe2wco_I

Did a bunch of work on the max msp patch and the arduino code. The arduino reads the ultrasonic sensor without delays. Max and arduino are now "handshaking", or exchanging messages in a call-response fashion. I might have lost a bit of hammer speed since adding the handshaking but it's worth having two-way communication.
With the communication working, the goal of this session is to think more about sound. I plug in a microphone and start experimenting with treating the sound from the hammers. Looking through my files for inspiration I plugged in a patch we made in class in CART 346, which is a combination of delays and filters. With a little tweaking, it transforms the hammer sounds into splash-like sounds. It's accidental but it feels interestingly related to my initial idea of making a piece for a playground jet structure.

Data from the ultrasonic sensor is also useable as numeric input within max/msp. I spend some time making first experiments with controlling different aspects of the sound treatment using the sensor. It's probably not enough - in the next prototype I want to use sensors to trigger different hammer motions as well. Maybe muliple ultrasonic sensors. Maybe I could try different sensors now that I have a working system to send data back and forth to max.