The Dance Dance Bot

Kaden Ward

200373058

CS 207

Trevor Tomesh

April 11, 2017

Table of Content

Introduction to the Dance Dance Bot

The project that I decided to create for my project in this class was a robot that plays the Wii game Dance Dance Revolution. This was simply a machine that would 'dance' with the player's input so that they could sit on the couch and protect those hard earned calories. Granted, you could always just play with the controller, but where is the fun in that?

The design of the Dance Dance Bot (The DDB for short) is four pneumatic pistons mounted on a stationary body that sits atop the dance mat controller. The player controls the robot with an infrared remote that is then translated through the Arduino UNO device to power a relay, and open a valve that extends a piston, pressing the dance pad. The pad will think that it is a human stepping on it, and will trigger the arrow as hit. The project's body will be built out of pure wood, while the pistons and valves will all be mounted with screws.

I don't believe that this has ever been done before; at least not made public on the internet. So I had to do everything with my own ideas and help from Trevor Tomesh. There has been some changes to my project's idea from what was submitted in the proposal. Mainly, the robot is not autonomous as was originally planned. I believe I was too ambitious with that idea, and ran out of time while making it to be able to give the robot eyes and the ability to see the arrows and decide what to press. Perhaps that will be the future of this project, but as for this class, it will remain purely human controlled.

Motivation

My motivation for this project was purely a 'spur of the moment' idea. I was contemplating ideas in my head when suddenly the inspiration to create a robot that plays Dance Dance Revolution hit me. This project is very unique as nothing to my research has been done to create a robot that plays this game. Other people have made and programmed robots to play other video games, specifically for speed running (called tool-assisted speed running), but nothing has been done before to play a dance game like this one. My project is also different than the speed run robots, as they usually use a microcontroller or PC to emulate a controller input. My project physically presses the buttons to play the game.

The main source of building instructions came from my background knowledge with pneumatics and the Arduino, as well as various online tutorials (included in References) and the help from Trevor Tomesh. Everything bought and installed was done so on a trial-error basis. Luckily for me, Princess Auto, where I bought most of my hardware from, is very good with returns.

Design process

The design of the DDB is ingeniously simple. It consists of four pneumatic pistons affixed to a wooden body, four pneumatic valves, an eight-channel relay (although a four channel will work as well if not better), a computer power supply, an infrared receiver, and any infrared remote. (A photograph of the project in a 3D design state can be located in Appendix A)

The body is made out of four wooden 'legs' attached underneath a large wooden square. Then four wooden sides are screwed in place along the sides of the square to make an open box with legs. The pistons were attached along the sides, one on each side, using 'Strap-it', and face down. They barely touch the ground when fully extended. This is an exact height, as too high up will not trigger the gamepad, and too low will lift the entire robot off the ground. Mounted atop the body is the four valves on each side. These need no special location or orientation as long as you can mentally keep track of the pistons. The Arduino is to be either attached or set atop the body, taking input from a camera looking at the screen, and sending it to the valves. The camera was later entirely removed as it was no longer needed for a man controlled gadget. Everything is to stand on the game's signature dance pad.

The body's design is designed so that the DDB's legs can hover over the directional buttons while not pressing any other button. This design is flawed however, as the pad has an A and B button in the top corners, where the two front legs are supposed to be resting. Angled legs would probably work better for this specific pad, but I got it to work by just moving the bot back a few inches. The resting legs were off of the center then, and the directions still gave correct input.

Many of the components are unspecified and interchangeable. The piston length does not matter as long as it can reach the 'floor' from the top of the body. Therefore, the piston length relies on the length of the legs, and vice versa. A metal frame could also be used, but wood is generally lighter, cheaper to purchase, and easier to cut and work with. Hydraulics is also a valid replacement for the pneumatics system, so long as you have the correct tools to use hydraulics.

Build Process

The building process is fairly simple once you get the hang of what you are doing. The prototype build is a simple button input - LED output system to get a basic understanding of how I'm going to handle the input and output of the system. First I attached four push-buttons to a breadboard. Next, I connected 5V power to one side of the button, and a 10K resistor to ground on the same 'side' of the button. Next I connect the opposing side of the button to analog in (I had trouble with digital-in consistency). Once I got the button input working, I attached four LEDs to another breadboard. Then I connect four digital pins to the positive end of the LEDs and connected a resistor to ground for each of the LEDs. This gave me an array of LEDs that I could control with button input (you can see this in Appendix B). This system became the foundation of the rest of the project, and everything evolved from there.

The next step after getting the LED-button system working was to build the body. I simply took some old 2/2 wood that I had lying around at my parent's farm and screwed them along the sides of an old square backboard. It didn't look great, but it worked like a charm, and most importantly, was free. After examining the pistons I was going to use, I decided to add a wood rail along the bottom of the chassis to attach to. This was again, old 2/4 that I had found on the farm, trimmed to size, and attached to the sides with screws. The body ended up being heavy, but it worked perfectly for my purposes. As mentioned above, the exact size of the body is a very loose criteria, and can be modified for smaller pistons and/or smaller dance mats. The only requirement is that the center of each side should hover over the arrows on the pad you are going to use.

Next came the pneumatics system. I had purchased (and would recommend you do the same) the pneumatic cylinders and valves one at a time, and attached them likewise. I tested each cylinder and valve before even considering how they would be mounted to the body. After each piston and valve was confirmed to work correctly, I would measure the center of a side on the body, and drill a hole there. Next I would attach about 5 inches of 'Strap-it' to the piston, then bend it into a brace shape over the side of the base. After extending the piston fully and marking where it would sit (how high it reached when extended and on the ground), I mounted the braced piston using basic screws. After mounting all four, one on each side, I set the valves on each side respectively, and nailed them into the base. I connected all the valves using a 4-way air splitter, and attached each valve's ports to the respective piston. After firing up the compression and checking everything was working, I moved on to what was surprisingly the most difficult part of my project, getting the power to work.

Now I'll spare all the failure builds that happened (they are all in detail in the failures and setbacks section) and get right into the one that finally worked. How I finally got the power to work was by using a relay system and a computer power supply. Trevor Tomesh helped greatly with this problem, and even loaned me a computer power supply to use for this project. After seeing him for help and receiving an 8-channel relay online, I began wiring the power supply to the relay and attaching the signal pins to the Arduino system, running a simple blink sketch on all four gates (I only used 4 of the 8 channels for the relay). Excited with this success, I quickly began work on an infrared input system.

The infrared input system is simply a hexadecimal input from a TV remote. I binded the specific codes for the left, right, up, and down buttons to the respective pins. The pins then open the relay gate and trigger the piston. This section of building was surprisingly easy, and can be bound to any infrared output signal.

Usage and Functions

The DDB is a simple gadget for the lazy, with a simple set up as well. All you need is your favorite TV remote! To set it up, grab your Arduino with the infrared system (see Appendix D) and upload the code in **infrared-input.ino** to your Arduino. When you open the serial monitor, you should see nothing. Press your desired 'Left' button, and make note of the number you receive. Do the same for Right, Up, and Down. Now open up **Project_V2.ino** and look for the variables Left, Right, Up, and Down at the start, and change those numbers to the ones you wrote down respectively. Now you should test to see if you entered everything correctly by seeing if your relay reacts to the pressing of the buttons.

Your pins should be as following: Infrared input to pin 5. Relay's 'UP' to pin 12. Relay's 'RIGHT' to pin 11. Relay's 'LEFT' to pin 10. Relay's 'DOWN' to pin 9. The system receives input from the IR remote, which starts a counter to allow the piston to stay down. While the counter is live, the Arduino writes high to the respective pin, which opens the relay gate, allowing the PC power supply to power the valve, extending the piston, pressing the button, winning the game. When the counter reaches 0, or after .65s, the power is off, and the valve releases the piston.

To play the game with the DDB, simply apply power to the PC power supply and Arduino, and supply air to the central air divider. If all is set up and binded correctly, you should be able to press a button on your remote and get a logical response. Now all you gotta do is start up your Dance Dance revolution game and set the project on the pad. You should be able to play fully from the comfort of your couch.

Special note, if the pistons don't seem to press the dance pad's buttons, place sheets of cardboard on the pads until it does. Be wary however, as too much cardboard will cause the robot to lift itself off the ground and move slightly. Simply remove some cardboard if this happens.

Setbacks and Failures

Most of the project was frustrating trial and error. The concept was fuzzy from the start, and with no other project to reference, I was left on my own to solve most of the problems. The bigger problems include getting the correct pistons and valves, getting the air compressor to work, getting the infrared system to work, and of course, the great power problem saga. The project was also littered with many smaller problems that took minimal effort to solve. Most of the setbacks timewise were simply waiting for parts to be delivered. In the end however, I managed to create a system that was relatively bug free and simple to adapt.

The first major problem was the piston and valves. At first, it was not so bad. The man at Princess Auto was very helpful in terms of what I needed and what I didn't need. I ended up returning to the store many times needing more, or different parts however. The valves were also sold out (they only had 3 in stock, and I bought them out) from the second time I went there. It took about 3 ½ weeks for them to come in aswell. In the end, this problem ended up being a simple solution as I knew what I needed to get on all my later trips, even if it was just a valve adapter or two.

The next big problem I encountered was getting the air compressor to work. This happened mostly in the time waiting for the valves to come in. The first air compressor I used was a very old thing that my grandfather had built in what looked like the 1800's. It worked fine for the first few tests, until it almost exploded. I use the word exploded loosely, as it really just had a small hole burst in the air tank. Needless to say, I could not use an air compressor that had a habit of breaking and leaking. This problem was solved fairly easily as well, as I just borrowed my parent's air compressor from their farm house. This one was newer, and never exploded during testing.

The third major problem I encountered while working on this project was getting the infrared system to work. This took place after the great power troubles, but I decided to save that for last. The problem I had with the infrared system was that the remote I was using had very little range, and was prone to giving different numbers every time. It was probably an encoded signal or something, but I never went too far into solving this problem. Another simple fix, I just switched remotes. After that, it worked like a charm.

Now for the epic tale of power. This all started with my lack of electrical knowledge. The valves I am using require 12V and at least .5A to trigger. I knew this, but had gotten it working with a 9V battery underpowered and thought that it would work. **This would not work.** So after seeing Trevor Tomesh for help, he loaned me a computer power supply and told me to use transistors. I did this, and had some failing results. At first, I was using small transistors, ones not designed for the 10A that are produced from the computer power supply. I ended up burning myself 3 times trying to take these transistors out. Next, I tried Darlington Transistors. These things can handle up to 25A and 40V, well within the range of what I was working with. Only problem with these Darlington transistors is that they need .1A to trigger. The Arduino didn't have enough amperage to trigger these transistors. After several frustrating attempts, I went back to Trevor, and he suggested using a relay system. Having everything to lose at this point, I quickly ordered an 8-channel relay from Amazon, being grateful for Prime 3 day shipping. I received the relay, and everything worked like a dream. I couldn't have solved this problem without the help I received from Trevor.

Some smaller problems I had were pistons being extended in the idle state (they should be retracted when idle). This was fixed by reversing the sides that were plugged into the relay. The pistons were also retracting too soon after extending. I solved this by increasing the delay counter in the code.

Milestones

This project in the end was a success. My goal was to be able to complete any one song on the easy difficulty, and I managed to complete 7 on easy. The project is not autonomous like was originally proposed, but that was a sacrifice I had to take to finish on time.

Everything else in the project went fairly smoothly, only encountering a few minor problems and couple major ones along the way. The body, pistons, and valves were all built exactly how I imagined them.

A major milestone in this project was getting the power system to work. As you have read in the failures and setbacks section, the power issues was the most difficult part of the whole project. Getting the system to work marked the time that I could finally begin working with input and infrared. Everything before was all done with physical test valves, a very inefficient way of testing the project.

Conclusion

In the end, I had an absolute blast building this project. I had never done anything on this scale before, and greatly enjoyed the experience of trial and error, as well as seeing my ideas and visions come to life in the physical world. At times it was frustrating and difficult, but that just made the successes even more enjoyable. I remember literally jumping with joy the first time I got the power working in this project.

I would have liked to get the Open Computer Vision system working, and make the project play the game all by itself, but as stated before, it was necessary to remove that component in order to finish by the deadline.

This project is far more simple that I originally planned, only requiring three different systems to work. The difficulty was not in building the systems, but getting them to work in harmony with one another. If I were to do this project again, I would change the body design. The pistons I used were unnecessarily large, and resulted in the ‘legs’ of the body to be unnecessarily large as well. While using a larger piston means easier to work with, the robot ended up using more air than necessary, only being able to do about three songs before requiring the air compressor to kick back in.

I believe this project taught me a large deal about not only electronics, but also computer power systems, pneumatics systems, and IR systems. This project also made me feel as if I was an inventor: researching my options, weighing the risks and rewards of each, and making a decision based that could change the entire project. While this project is never going to be a household device, nor will it ever be an industry standard of anything, it still made me feel as if I was creating the next big thing. I believe that is the feeling that we should take from CS 207. The feeling of being an inventor.

References

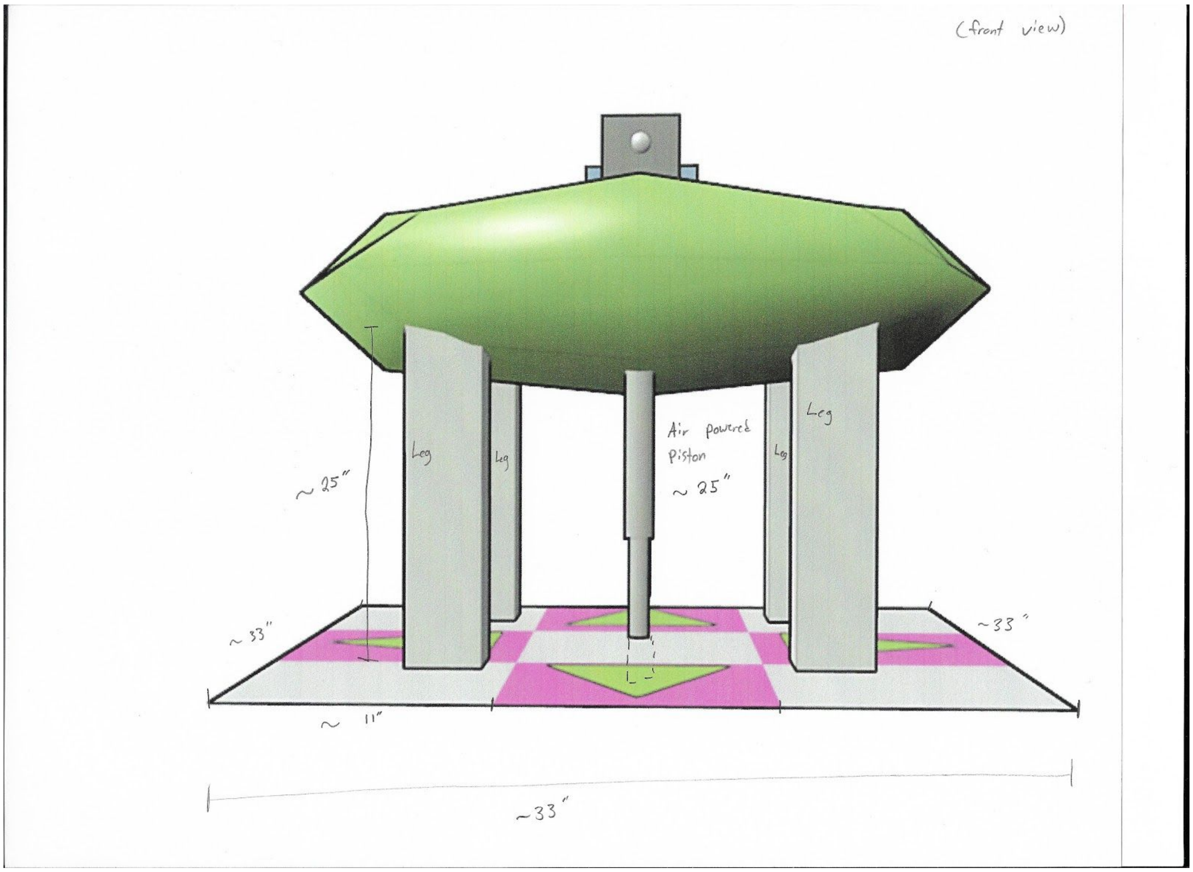
DIY Tryin. “Use Arduino to Switch Power On and Off.” *Youtube*. Youtube, Apr 21, 2014. Web. https://www.youtube.com/watch?v=tuOrH9sSykk

Alronzo. "IR Communication." IR Communication. Sparkfun, Date Unknown. https://learn.sparkfun.com/tutorials/ir-communication/discuss

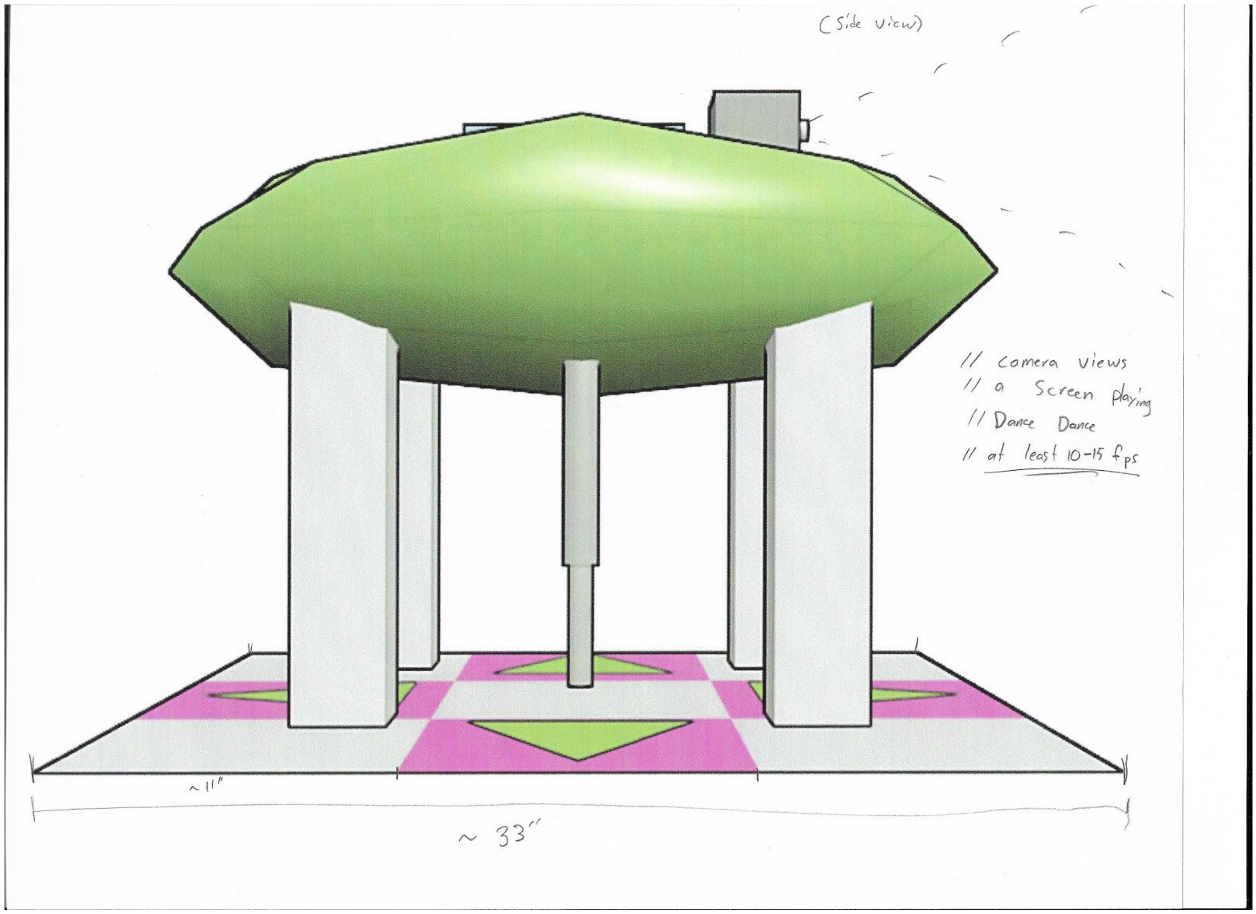
STCValves. “Solenoid Valve Electrical Connection Procedure.” *Youtube*. Youtube, jul 15, 2013. Web https://www.youtube.com/watch?v=_tby5VuvNuk

Appendix

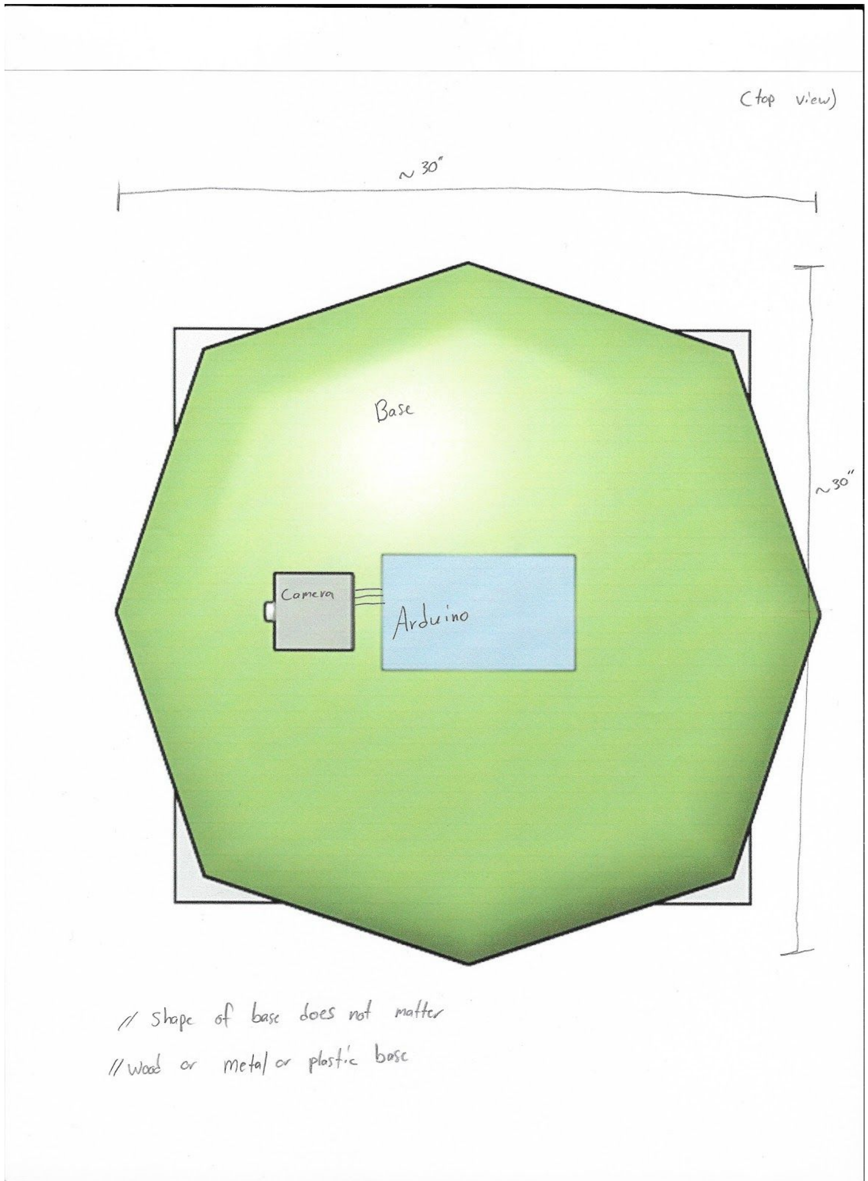
Appendix A: 3D models and design phase concept art.



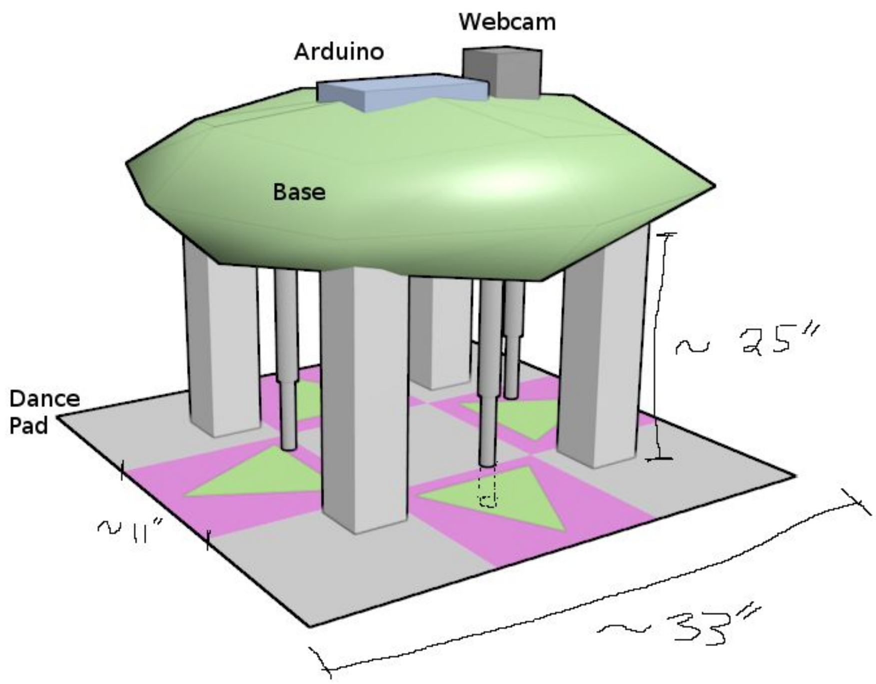
Front view of concept model with diagram markings.



Side view of concept model with diagram markings.

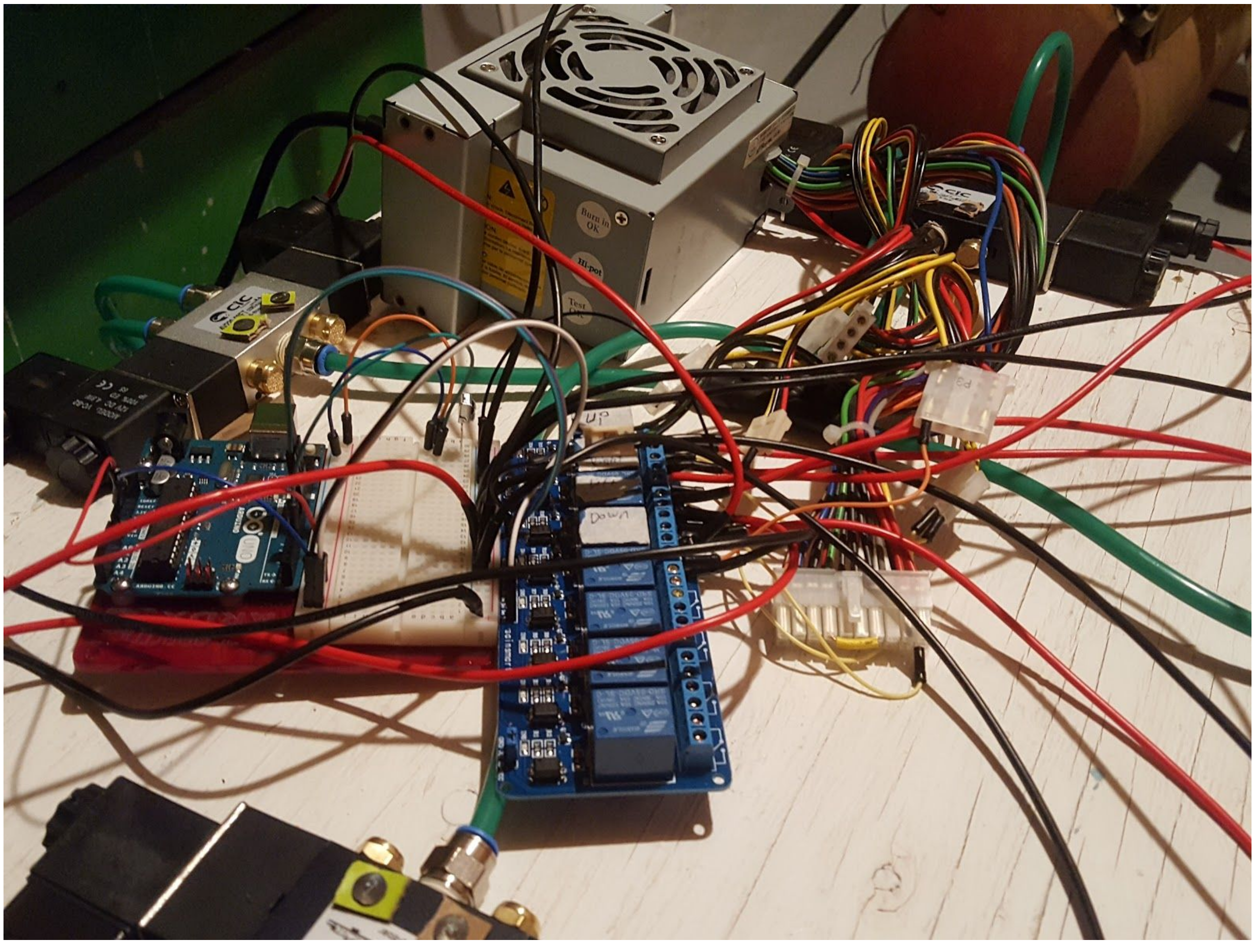


Top view of concept model with diagram markings.

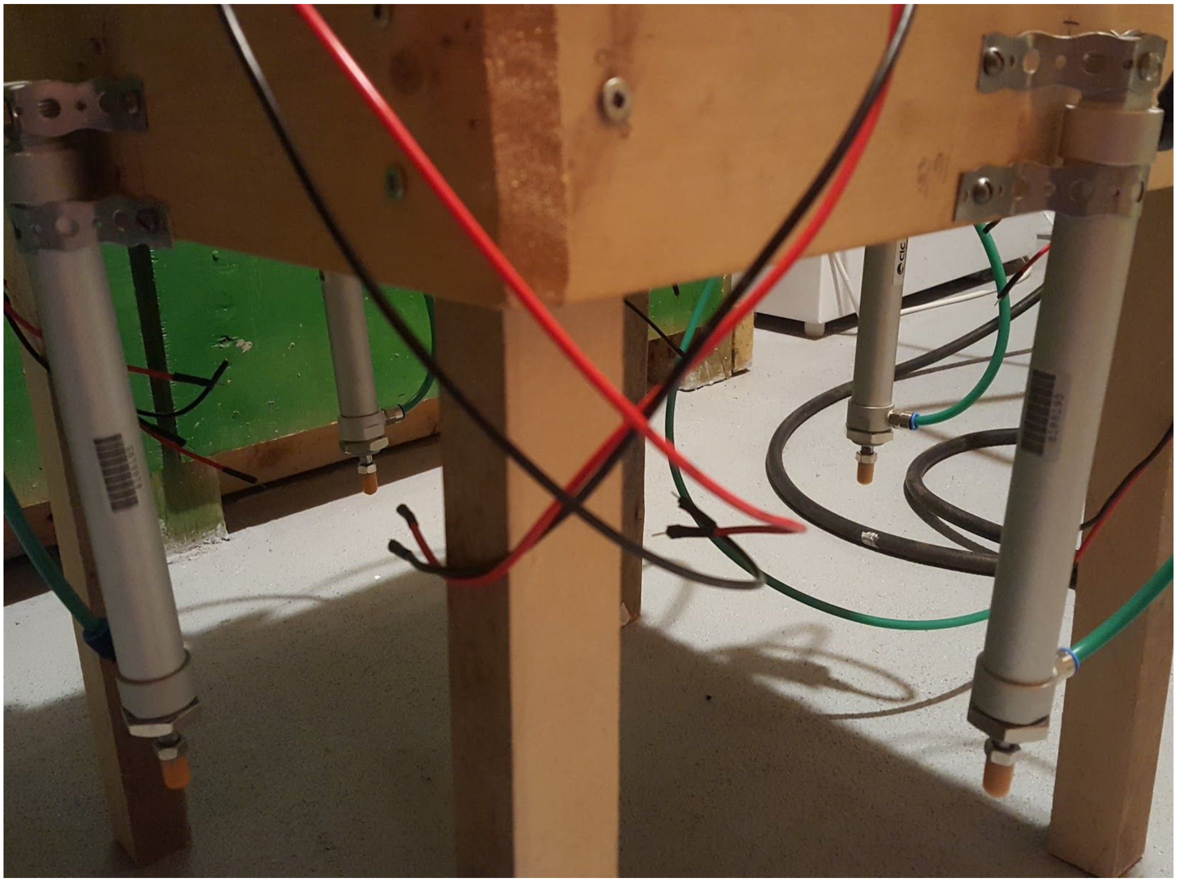


¾ view of concept model with diagram markings.

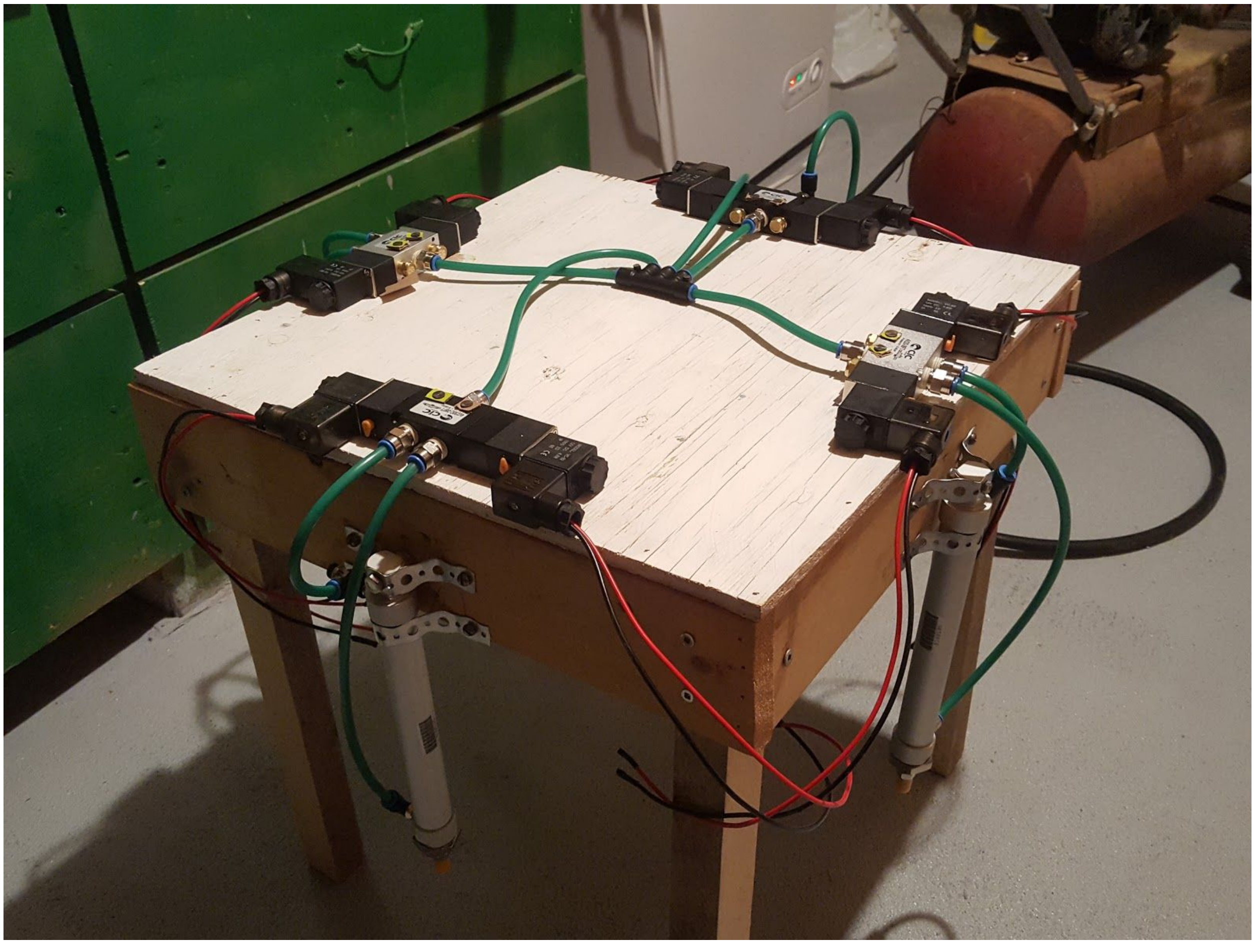
Appendix B: Final project images.



Top view of electronics of the final project state.



Leg view of pistons of the final project state.



¾ view of project without electronics.

Appendix C: Code locations.

All code for this project can be found in the **build** folder in the repository listed at:

https://github.com/ward208k/CS207_Project