

# Bubble Cannon Interactive Game Controller

*Design, Manufacture, and Implementation of an Interactive Aim and Fire Game*

## Team

- **Jimmy Huang** - Researched and implemented electrical hardware. Developed Arduino code to read in inputs from our sensors and output data to serial to be read by the Java program; assisted in physical device assembly and craftsmanship
- **Michelle Chang** - Modified the game's original Java code to read in the serial input from the sensors and map them as new controls for the game, assisted in physical device assembly, craftsmanship, and decoration
- **Serena Chang** - Designed and laser cut two iterations of the physical game enclosure and experimented with different user interactions with the physical controller, assisted in physical device assembly and craftsmanship.

**Final Project Video:** <https://www.youtube.com/watch?v=v3Hv8G64tA8>

**Prototyping Functional Test:** [https://www.youtube.com/watch?v=6IKVDIa\\_VLo](https://www.youtube.com/watch?v=6IKVDIa_VLo)

**Github Repo:** <https://github.com/idd-fall16/frozen-bubbles>



## Overview:

The game we chose was Frozen Bubble, a variation of the classic arcade game, Puzzle Bobble, and the Neopets game, Faerie Bubbles. The game involves aiming a cannon and firing differently-colored bubbles into a field of additional bubbles. If a fired bubble connects to form three or more bubbles of the same color, the bubbles get knocked off the playing field. However, additional bubbles are added to the playing field from the top of the screen in an accelerated manner. The player loses when the field of bubbles passes below an end line towards the bottom of the screen. Below are screenshots of the game:



We designed and manufactured a laser-cut cannon as a physical representation of the bubble-firing cannon in the game. Turning the physical cannon aims the virtual cannon and pulling a trigger on the cannon fires a bubble.

In terms of the physical appearance of the controller, we chose to design the base of the cannon in such a way as to provide ease of control and a more comfortable grip. As such, our cannon rests on top of a sizeable piece of circular wood and has ergonomic handholds built into the steering pad.

The cannon requires two methods of input. A potentiometer in the base of the cannon outputs values that are mapped by software to produce absolute positions on the screen that determine how the virtual cannon is displayed. A joystick, our source of discrete input, is mounted within the cannon and acts as our firing switch (the original Frozen Bubble game mapped from key-up to fire, instead). When pulled, the virtual cannon will release a bubble on screen.

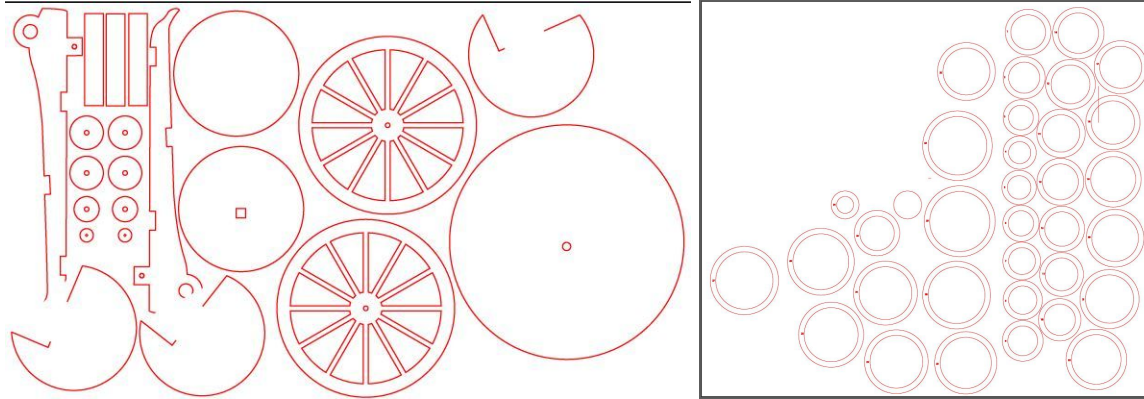
## Process and Implementation

We first began by brainstorming different ways to implement rotational motion to change the cannon angle and discrete motion to fire a bubble. To do this we first had to understand the implementation details of various sensors to decide which sensors would be most useful for our controller. Some of our concept ideas can be seen below:

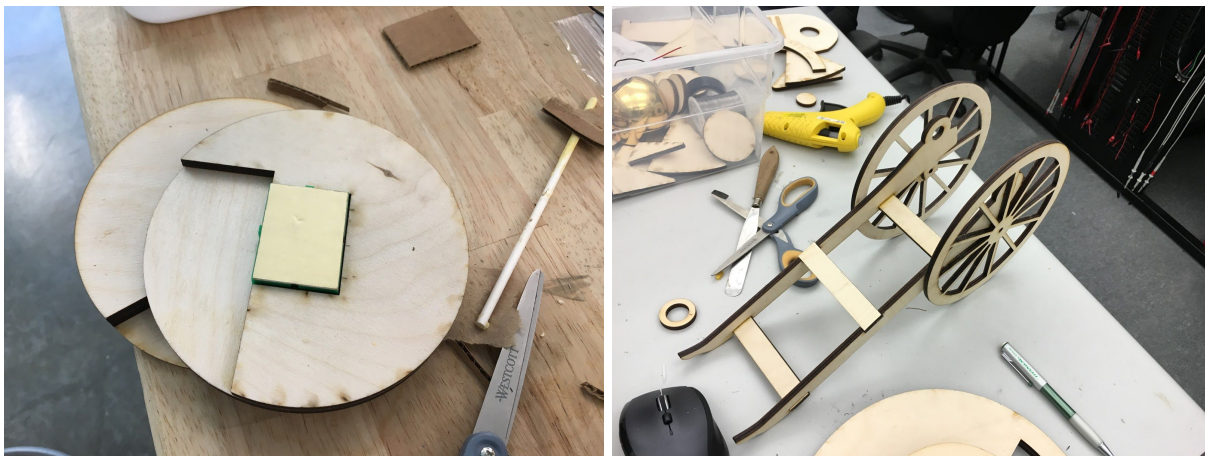


## Design and Manufacturing

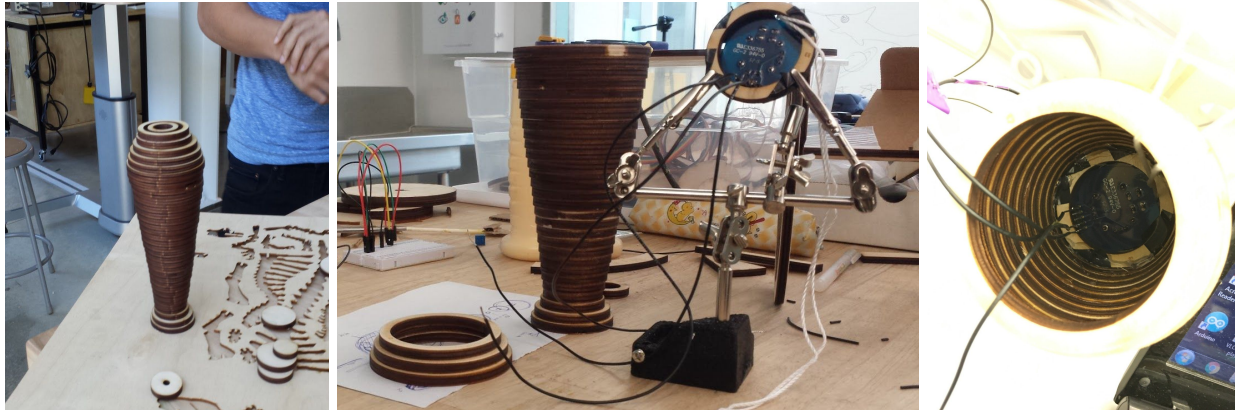
Our manufacturing method consisted of laser cutting three (3) main components: the base, chassis, and cannon. Outlines of our 2D laser cut designs can be seen below:



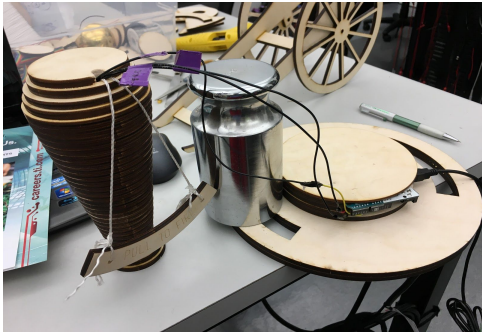
Our base was constructed of 4 layers of  $\frac{1}{4}$  inch wood with square geometries cut out for placement of the electrical components and microcontroller. Meanwhile the chassis was constructed using 2 parallel support structures with 3 perpendicular ribs of identical thickness.



The cannon and joystick firing mechanism was a bit more involved to manufacture. The cannon itself is actually a series 30+ individually laser cut rings of various diameter stacked on top of one another to create the outer profile of a cannon. The joystick firing mechanism was mounted inside the cannon by first soldering the electrical components and trigger mechanism and then mounting onto a specially designed wooden ring. This mounting ring was then attached to the inside of the cannon by perfectly fitting with the inner diameter of one of the previous rings.



The end cap of the cannon was created using a solid disk rather than a ring with a notch cut out on one corner to allow the wires and firing mechanism to pass through.



Finally the three assemblies were painted using watercolor. The intention behind using watercolor instead of traditional oil based paints was to maintain out the natural aesthetics and grains of the wood. In addition, we tried to keep to the theme of our “frozen” bubble cannon game by painting traditional winter images.



## Finished Project

- Final Project Video: <https://www.youtube.com/watch?v=v3Hv8G64tA8>



## Reflection

### The Physical Device

The final product is very intuitive, as it is instantly clear to the user what to do with the cannon and accompanying fuse (the added signage also helps in this regard). However, some pain points that came up while testing our device include a clunky user experience, an insufficiently-large base, and continuous firing errors regarding reading in discrete input.

We envisioned that users would interact with the device by turning the cannon and firing, like they would in the game; essentially, the game mechanics brought to life. In practice, however, this process is a bit clunky. While the steering board is designed with two handles for the user to grip and steer with, in reality the process is quite quick. Users want to be able to turn the cannon fast so they can then pull the trigger. The reality is that there is no need for two handles, as players frequently opt to leave only one hand on the steering wheel, while the other grips the trigger. However, because the firing handle is attached to the base of the cannon, turning the cannon sometimes puts the user in an awkward position because they can no longer pull straight back to fire. Instead, they must yank with their arm at, for instance, a 45-degree angle. Future improvements could, perhaps, look into separating the firing mechanism from the steering mechanism so that one does not affect how the player uses another.

Another issue we discovered is that the base of our device is much too small and light to provide enough friction for the steering wheel. As a result, if merely placed on a table without some method of securing the device (such as tape), in turning the steering wheel, the user may end up turning the entire game controller instead of just the cannon. To solve this, future iterations could look at building a heavier base board for the device, or adding a padding of rubber (to increase friction) to the underside of the base.

Finally, despite our best efforts, consecutive firing errors do still occur occasionally. This happens most often at the very beginning of the game, between the splash screen and the actual game screen. Future iterations could, perhaps, experiment with adding a slight delay when the game screen proper is loaded.

### The Process

On the software side, the hardest part was dealing with delay issues (we approached this with more of a trial-and-error technique) and trying to figure out how to read in from serial. Aside from the gruelling process of installing the proper dependencies to make RXTX work as intended and trying to figure out the logic behind previously written open source code, the process was fairly straightforward.

On the hardware side, the most challenging portion was mapping the potentiometer to the angles. The potentiometer we began with was a trimmer pot, so the values it returned were not consistent. We had to trim the full range of motion of the potentiometer (approx 270 degrees of rotation) and then map it to an appropriate angle in the game. There were many issues with this calibration. The readings were inconsistent from time to time, so it was difficult to get a reliable measurement. We tried using a number of different pots in trying to solve this problem, and finally settled on purchasing a higher end potentiometer.