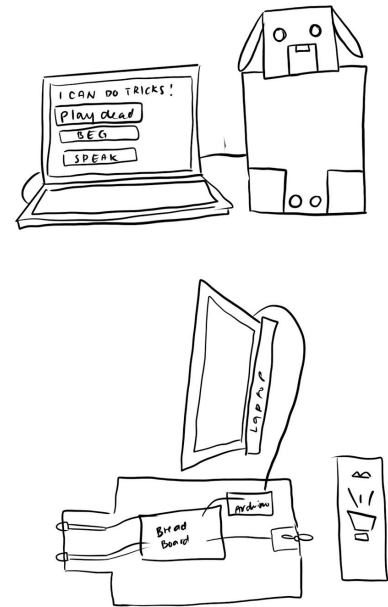


Quim Paredes

Concept:

Building on my dog for my midterm, I want to create a virtual dog that you can interact with. It will have all the same features as the original (sensing when there is a person and wagging its tail, wagging its tail faster when you pet it. But I will be adding barking and whining audio through processing. In addition, you will be able to make the dog do tricks. It will speak (bark), play dead (LEDs off and tail stops wagging) and beg (whining). The user can interact with my clicking the commands on my laptop. I also hope to add the feature of when you touch the dog's bone in its bowl, it's eyes will flash red, the tail will stop wagging and it will start growling. For this I will use a light sensor. In addition, I will make the casing for the wiring more aesthetically similar to a dog.



Arduino Program:

I'll be using the last program I made as a basis. I will send a certain character to Processing when it senses a person, making it bark. In addition, it will send a character when the light sensor senses that it is not covered (by the bone anymore) and, make the dog growl and activate the red LEDs. In addition, it will communicate with Processing to see when a certain command is clicked to do the trick (play dead, beg and speak).

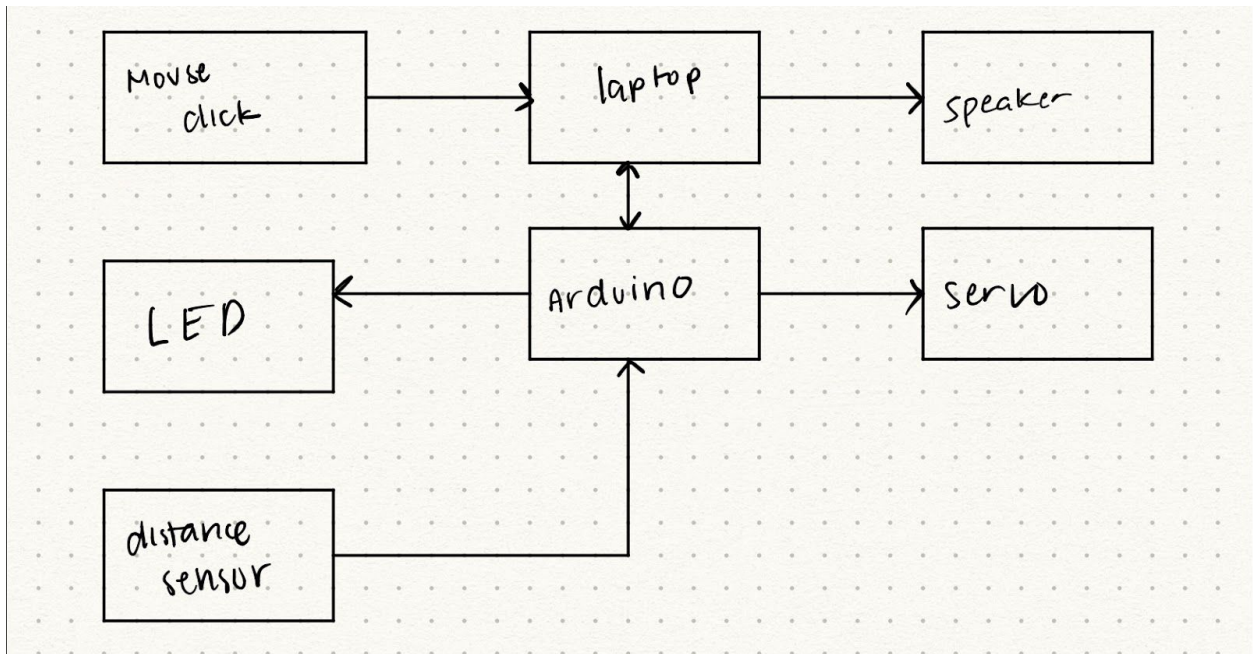
Processing Program:

The processing will be fairly straight forward. I will use the mouse click function to decide what trick the dog should do. And by default, the dog should be barking.

Technical Requirements:

- Learn how to display audio from processing
- Learn the mouse click function and make graphics

Block Diagram:



Equipment Requirements:

- Servo motor
- 2 light sensors
- Yellow LED
- Red LED
- Laptop
- Faux fur or yarn (gray, black and brown)

Areas of Greatest Concern:

I'm most worried about using the servo motor again and learning how to use the sound function in Processing. Also, because I'll be working with `millis()`, I'm worried that I won't fully comprehend the program enough to run it smoothly.