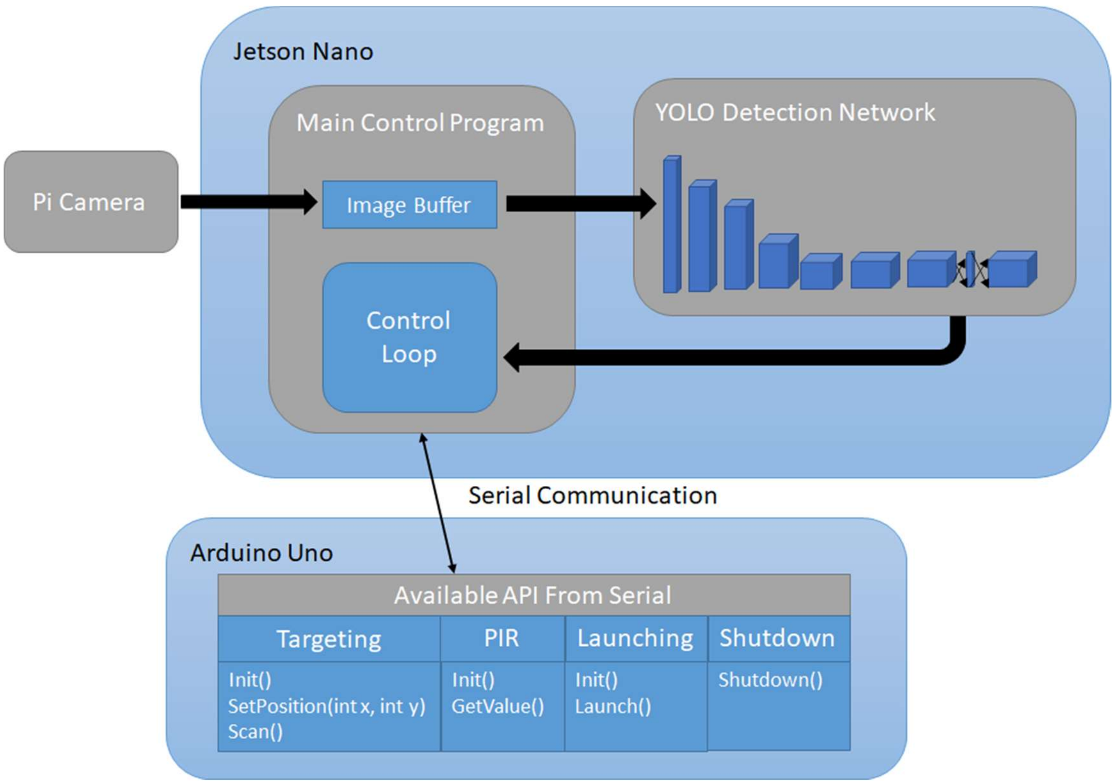
Final Report

Revised Project Design



Above is our final software architecture. The Jetson Nano will be used as the main controller, and the arduino will be controlled by the Jetson through an API that is available on the serial channel between the two devices.

Targeting:

Init() - setup GPIO pins

SetPositon(int x, int y) - moves the motors to the position (x,y), if scanning stops the scan.

Scan() - Turns 1 degree every 200 milliseconds for 180 degrees. Once completed it will reset back to the position (0,0)

PIR:

Init() - setup GPIO pins

GetValue() - returns the value of the PIR sensor

Launching:

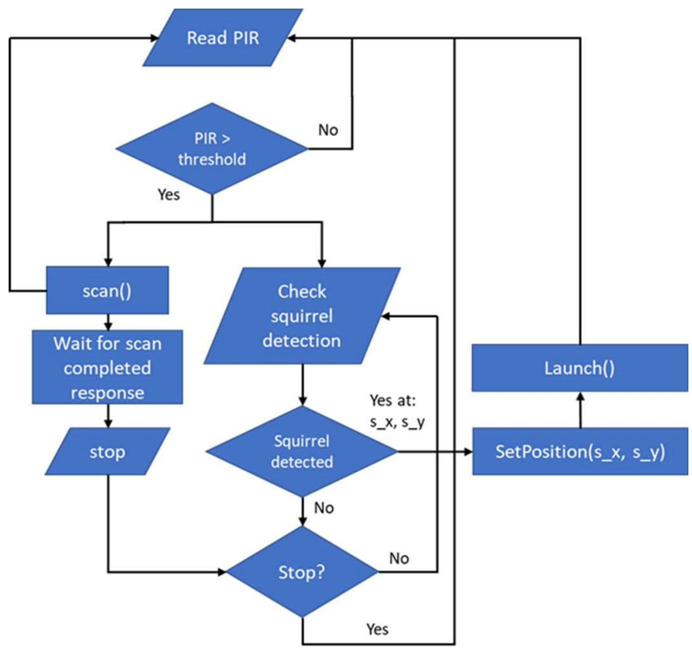
Init() - setup GPIO pins

Launch() - Will launch the ping pong ball and reel it back in

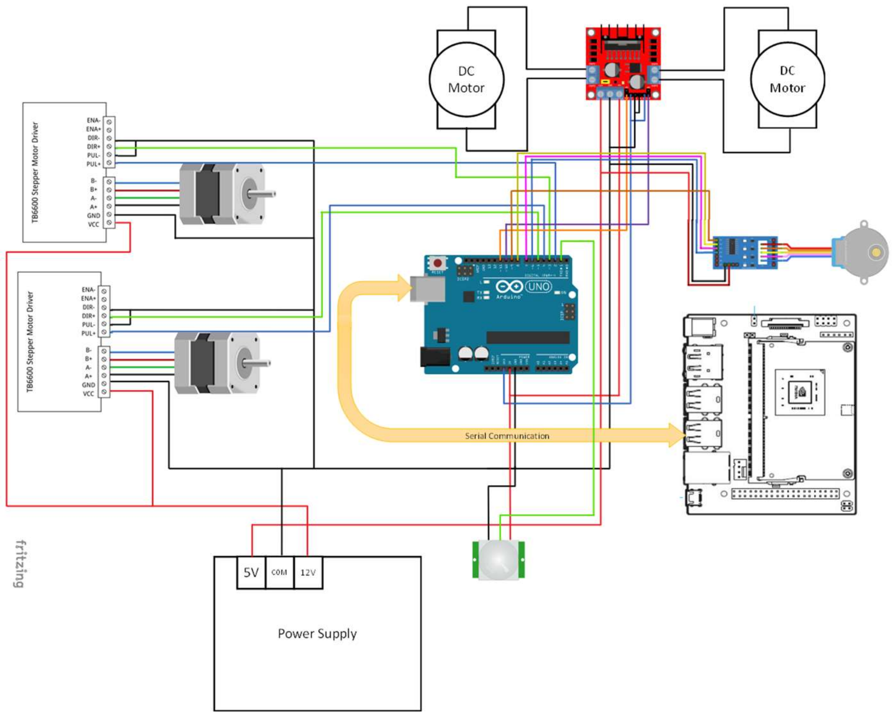
Shutdown:

Shutdown() - resets all GPIO pins on the arduino before it is shut down so that no unexpected behavior occurs on bootup

Launching is controlled by the main control loop. You can see a block diagram of the control flow for this loop below. When the PIR value is greater than the threshold for movement the camera begins to feed images into the image buffer, which are in turn fed into the YOLO detection network. If the YOLO detection network detects a squirrel it sends the coordinates from the image to the control loop which then decides where to move the motors to target the squirrel.



Implementation Details

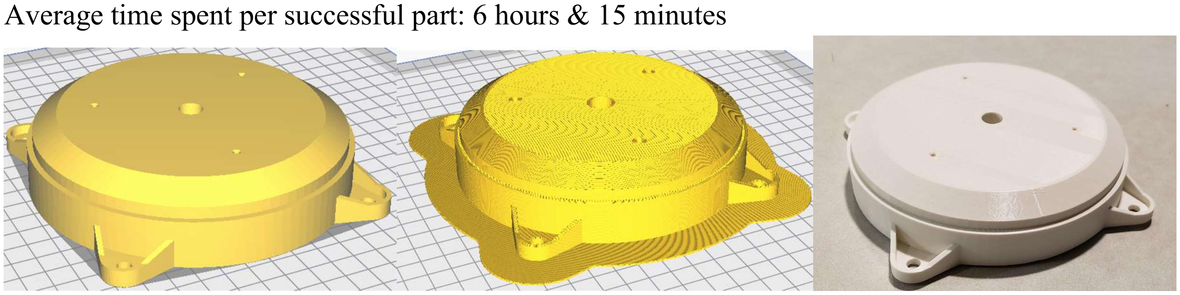


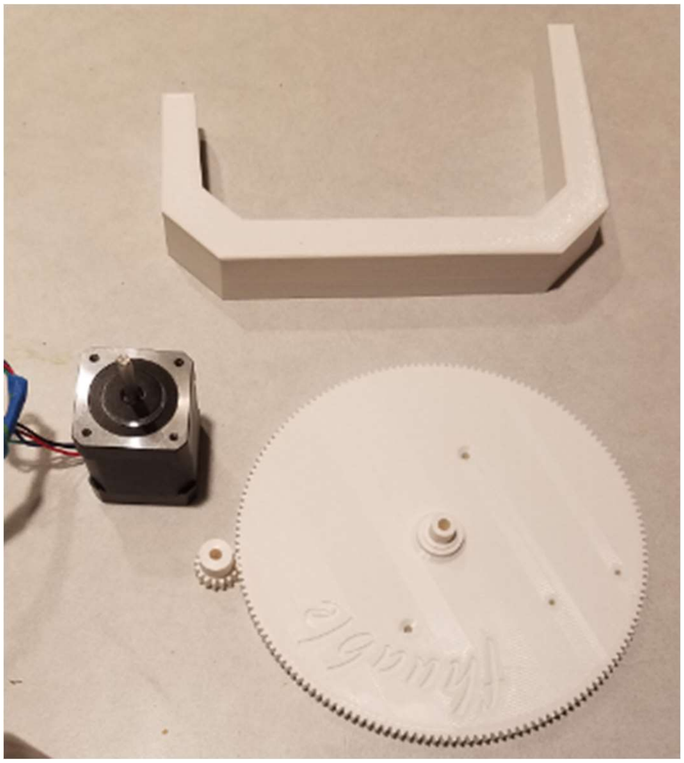
Above is the wiring schematic for the turret. In future versions of this product this could be turned into a single PCB board that could be used as a shield that you attach to the arduino.

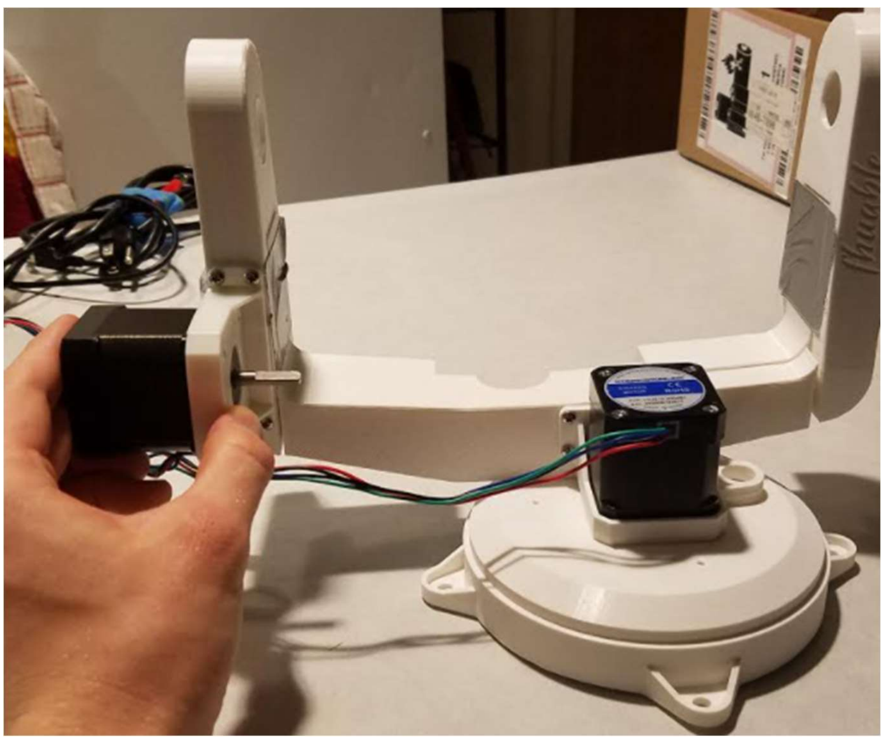
3D printed parts

There were 27 printed parts of different sizes and print times. There were initially warping errors when attempting to print the large parts. To debug we reprinted with ABS and PLA at varying nozzle and bed temperatures.

Total plastic used: ~1.4kg
Total time spent for only successful parts: ~167 hours
Average time spent per successful part: 6 hours & 15 minutes







Testing Process

A - Functional Testing

The following is a summary of our testing results, all tests were done manually:

Passing = (green) Failed = (pink) Mixed = (yellow) No Test = (crossed box)

	Subsystem				
Test ID	Targeting	Launching	Reeling	Vision	Power
S1	Passing	Passing	Failed	Failed	No Test
S2	Passing	Passing	Failed	Failed	No Test
S3	Passing	Passing	Failed	Failed	No Test
S4	Mixed	Passing	Failed	Passing	No Test
S5	Failed	Failed	Failed	Mixed	No Test
S6	No Test	No Test	Failed	Mixed	No Test
H1		Failed	Failed	Failed	Failed
H2	Failed	Passing	Failed	Failed	Mixed
H3	No Test	Passing	Failed	No Test	No Test
H4	No Test	No Test	Failed	No Test	No Test

Functional tests are detailed in Section 5.3 of our design document in appendix VI. Here “S1” refers to Software Test 1, and “H1” refers to Hardware Test 1.

B - Detail of Failing Functional Tests

Targeting:
(S4) - Fails to set pitch and yaw when shown a squirrel image. This is because the computer vision pipeline has not yet been integrated with the targeting system
(H2) - Fails, also due to integration issues.

Launching:
(S5) - Test is based on integration of the vision system which we have not completed and thus it fails.
(H1) - The motors do not spin fast enough to launch the ping pong ball due to a power issue.

Reeling:
All Tests - We have not done any work on the reeling system

Vision:
(S1-S3) - Test is based on PIR sensor. We have yet to connect the PIR sensor.
(**S5-S6) -** Untested
(H1-H2) - Test is based on PIR sensor. We have yet to connect the PIR sensor.

Power:
(H1) - Our power supply is not sufficient. We are able to power all parts besides the flywheel DC motors which, with our current design, can only be connected to a 5V @ 4A supply, however we need a 6-12 V @ >= 3 A supply to fully power the motors.
(H2) - Voltage at all nodes is as expected besides the voltage at the L298N driver used to drive the flywheel.

A - Non-Functional Testing

We have not yet completed any non functional test besides **(5)** and **(6)** from section 5.4 in Appendix V. Both these tests pass. The other tests required the system to be completed to do any of the tests, and we just recently finished the vision system and have not had the time to integrate it with the launching and targeting systems.

Context of Related Products and Literature

There are currently no products on the market like ours that attempt to deter squirrels by firing projectiles at them. Almost all squirrel deterrents are some form of scented chemical spray.

Our Squirrel detection system is not unique in that it uses the popular YOLO Deep Neural Network to do the detection.

See section 3.1 of appendix VI for more detailed information, and other projects similar to ours.

Appendix I: Operation Manual

In its current state the application is incomplete and requires manual control to do testing.

To launch the squirrel detection system on the Jetson open the terminal and enter the following commands:

“python3 -m venv env”
“source env/bin/activate”

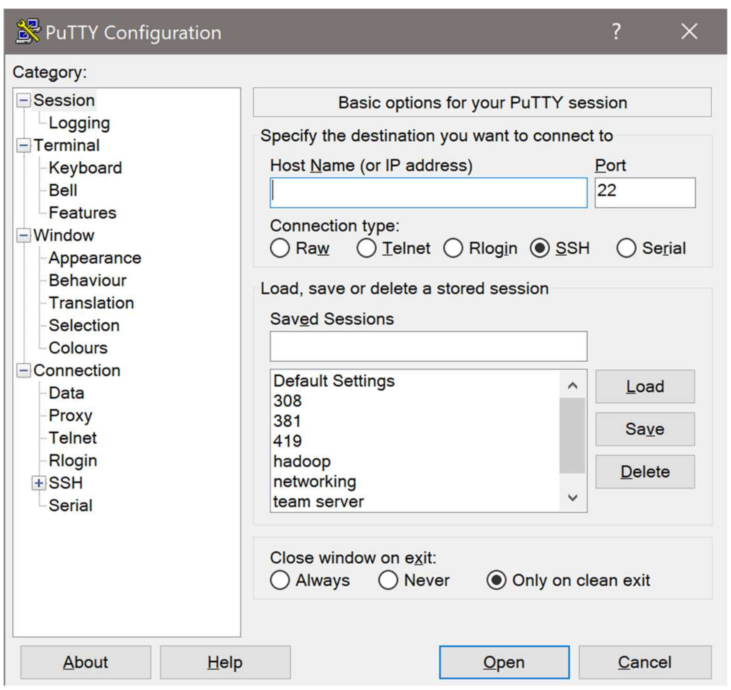
“export LD_PRELOAD=/usr/lib/aarch64-linux-gnu/libgomp.so.1”
“sudo service nvargus-daemon restart”
“cd Squirrel_Detector/”
“cd 3_Inference/”
“Python3 Detector.py --is_tiny”

After inputting the last line, the program will start running. After 30 seconds, the camera will display a live feed, with boxes drawn around detected squirrels. If a squirrel is detected, it will also print the coordinates of the center of the squirrel. To stop running, click ‘ctrl c’. Then “sudo service nvargus-daemon restart” to make sure the camera is correctly freed up.

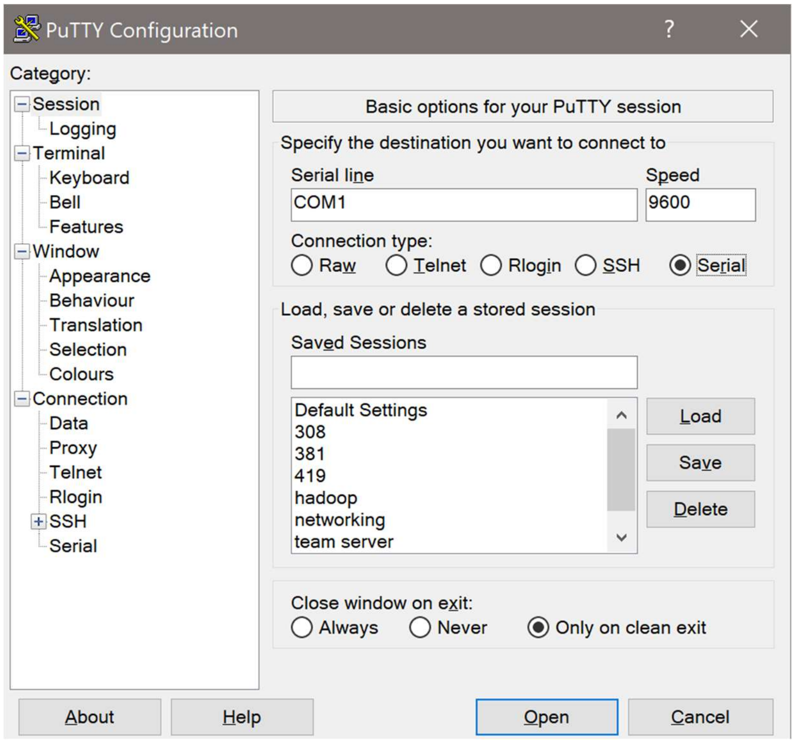
The launching and targeting can be controlled through a computer or the jetson by connecting to the USB output from the arduino, and using a serial communication tool such as putty.

For connecting to serial from Putty:

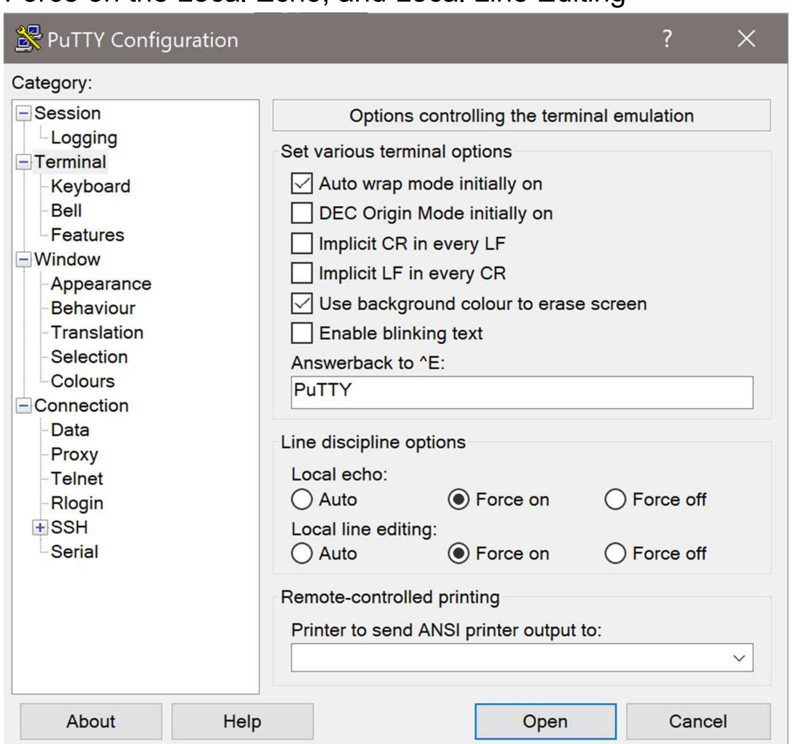
Open putty



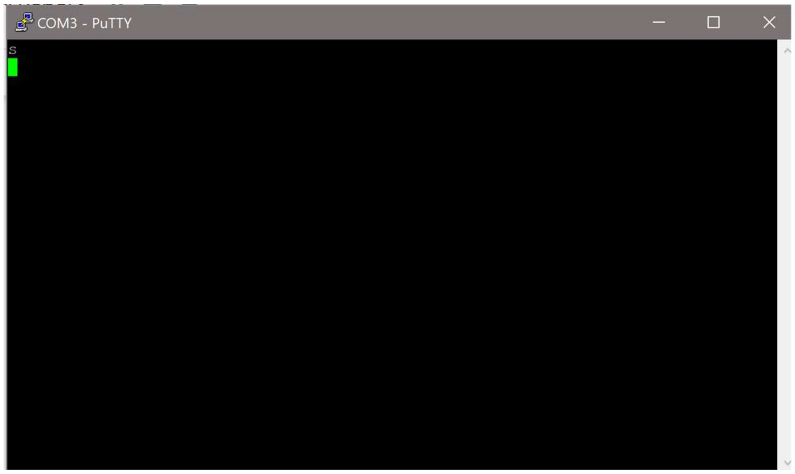
Select serial and set the COM port to the port that the arduino is connected to



Force on the Local Echo, and Local Line Editing



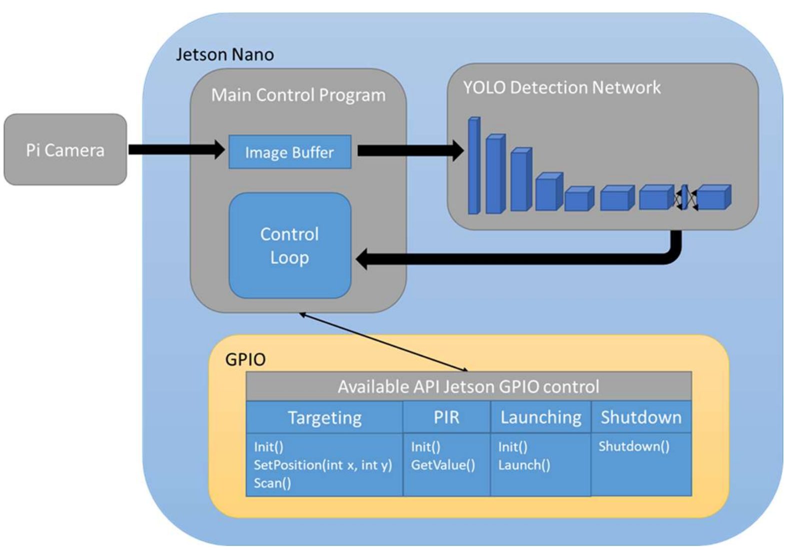
Click open



You will see a terminal window. There are two operations:
S - scans.
P - stops the scan, launches the ping pong ball, and sets the motor positions back to home

Appendix II: Alternative Versions

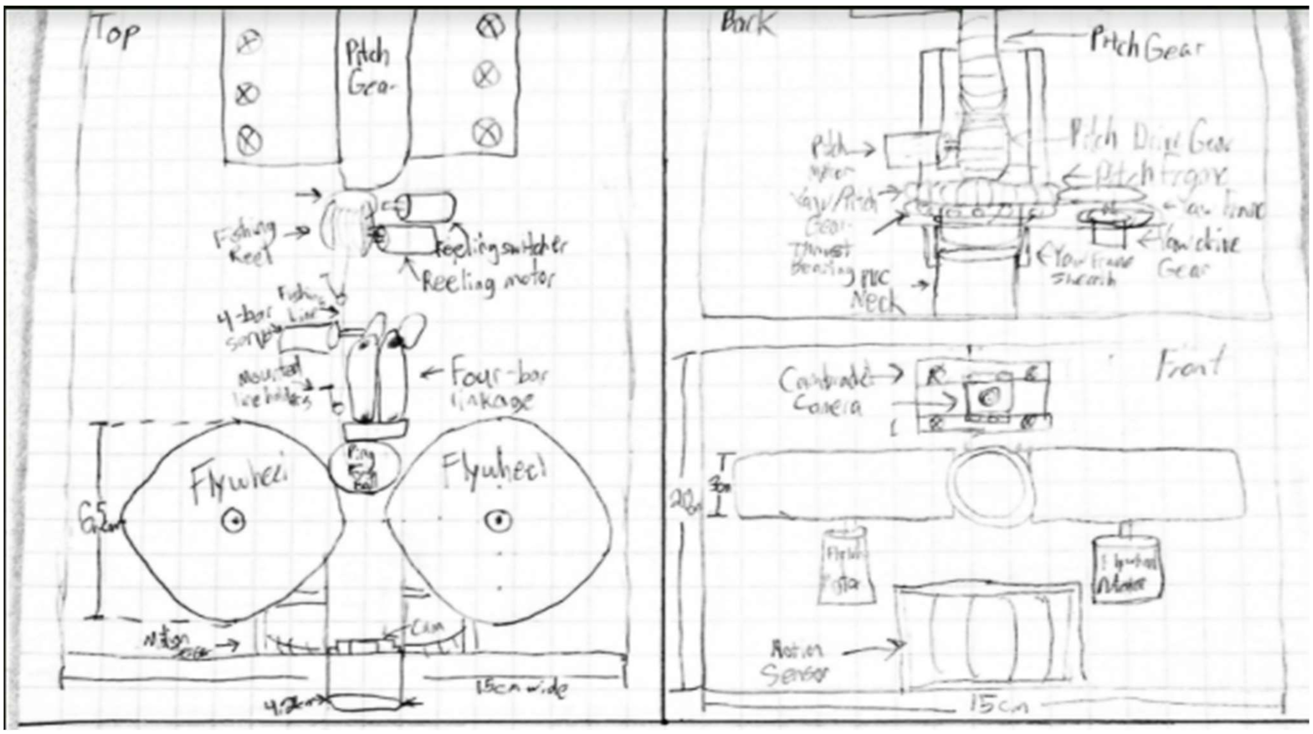
A - Initial Program Architecture

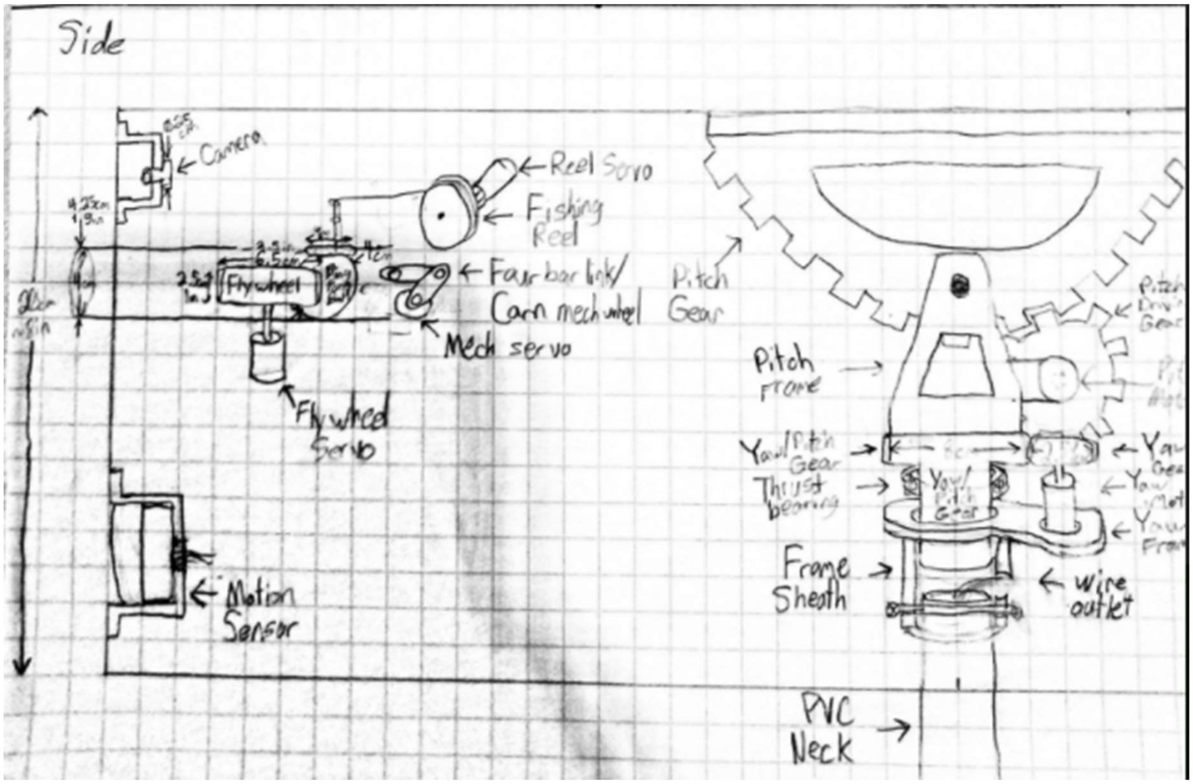


In our original design we planned to use the GPIO pins on the Jetson nano to control the turret and read from the PIR sensor. We initially believed that this would be possible as the output voltage of the GPIO pins is documented as 3.3V, but when we attached them to our motor drivers the voltage dropped to 1.5V. We were unable to figure out the cause of this so we decided to use an arduino uno instead for motor control.

B - Custom Targeting System

We originally had designed a custom targeting system, but we had issues creating the models to 3-D print as we are not mechanical engineers and did not know what we were doing. We had looked last semester for pre-made turret systems that we could 3-D print but were unable to find any. However, after looking again this semester, we were able to find a gimbal system on Thingiverse by Fhuable.





C - Displaying notifications with an LCD

Initially we had an idea to implement a notification system through the use of an LCD. This would allow us to let the user know when issues occurred and then give insight about resolving those problems. It could also display stats or messages other than errors that would be of interest to the user. The skeleton code is mostly implemented but untested. The LCD would be one of the final steps to complete our original design, and we didn't have time to get to it due mostly to COVID-19 restrictions.

Appendix III: Points Worth Mentioning

A - Unit Cost

Part	Quantity	Cost
Jetson	1	$99
StepD-01	1	$14.95
Pi Camera	1	$29.33
Nano Power Supply	1	$7.50
SSD 64 GB	1	$14.99
Nema 17 Steppers	2	$27.98
DC Motors	2	$13.78
L298N H-Brige	1	$5.99
TB6600 Driver	2	$23.98
6 ' Power Cord	1	$6.16
3D-printing plastic	2	$40
Casing	1	$15
Power Supply	1	$20.77
Wirering	1	$20
Arduino	1	$15
Total		$354

A big portion of the project is the ability to keep the whole deck safe. Because of this we needed to have a system that could be deployed at multiple locations on the deck. This means that it needs to be low cost so that we can have more than one of them, but the cost will be low enough that our solution is competitive.

Appendix IV: Code

```
void loop() {

  currentMillis = millis();
  String command = String("");
  if(Serial.available() > 0)
  {
    while (Serial.available())
    {
      command = Serial.readString();
    }
    //Serial.println(command);
    if (command.charAt(0) == 's')
    {
      Serial.println(command);
      scanning = true;
      prevMillis = currentMillis;
      scan_ix = 0;

    }
    else if(command.charAt(0) == 'p')
    {
      char x = command.charAt(1);
      char y = command.charAt(2);
      int a = (int)x;
      int b = (int)y;
      Serial.println(a);
      Serial.println(b);
      scanning = false;
      set_position(x, y);
      launch();
      goHome();
    }
  }
  if(scanning && (currentMillis - prevMillis) > scan_delay)
  {
    step_forward_one_degree(pulPin_y);
    scan_ix++;
    prevMillis = currentMillis;
    if(scan_ix == 180)
    {
      scanning = false;
      delay(100);
      goHome();
      Serial.println("e");
    }
  }

}
```

Above is the main loop for the arduinos program.

Appendix V: Object Detection Accuracy

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	image	xmin	ymin	xmax	ymax	Detected xmin	Detected ymin	Detected xmax	Detected ymax	confidence	Detection Result	Intersection	Union	IoU
2	milan-gurung-nSoe5nE8Pq8-unsplash.jpg	none	none	none	none	718	1217	2718	4368	0.34378365	False Positive	N/a	N/a	N/a
3	felipe-bustillo-S-5_g_tvBGA-unsplash.jpg	none	none	none	none	none	none	none	none		True Negative	N/a	N/a	N/a
4	alex-7pTsJgiVlMg-unsplash.jpg	none	none	none	none	1123	1834	2853	4543	0.43954605	False Positive	N/a	N/a	N/a
5	cristina-schek-oJieg2n8duk-unsplash.jpg	4054	1881	5329	2922	3977	1804	5399	3137	0.6576522	True Positive	1326914	1895526	0.70
6	deepak-h-nath-iHOvGljFpSg-unsplash.jpg	1210	2314	2490	4452	none	none	none	none		False Negative	N/a	N/a	N/a
7	hillie-chan-kKsbR_BGwlE-unsplash.jpg	1926	1316	2526	1997	1797	1365	2643	2160	0.67516077	True Positive	379313	702191	0.54
8	melvin-thambi-5BlLwkk6w6I-unsplash.jpg	2044	635	3617	2249	none	none	none	none		False Negative	N/a	N/a	N/a
9	joakim-honkasalo-3Xdff7Amc8k-unsplash.jpg	1345	888	2429	1991	none	none	none	none		False Negative	N/a	N/a	N/a
10	mathew-schwartz-mRitYzw9_-I-unsplash.jpg	1055	314	3031	2344	919	0	3560	2531	0.44462132	True Positive	4011634	6684371	0.60
11	ilnur-kalimullin-D3fLgDc9uOQ-unsplash.jpg	1938	1147	3978	3183	1924	1122	4712	3069	0.72002935	True Positive	3920785	5661449	0.69
12	kulli-kittus-qyt0cPByJjs-unsplash.jpg	1350	686	2583	1956	none	none	none	none		False Negative	N/a	N/a	N/a
13	caleb-martin-Tk71SYS8UBY-unsplash.jpg	2231	805	4696	2801	2251	1124	4777	2578	0.89104784	True Positive	3555285	5035935	0.71
14	IMG_0502.jpg	1950	1932	2188	2258	1776	1823	2338	2277	0.77029121	True Positive	77481	255148	0.30
15	IMG_0515.jpg	363	1492	1705	2033	468	1517	1805	1992	0.69712096	True Positive	587640	773754	0.76
16	IMG_0716.jpg	930	1677	1883	2435	953	1559	1872	2522	0.66349566	True Positive	696324	910465	0.76
17	IMG_0743.jpg	2527	2097	2854	2341	2506	2081	2871	2323	0.65300775	True Positive	74066	94356	0.78
18	IMG_0743.jpg	363	2092	474	2300	none	none	none	none		False Negative	N/a	N/a	N/a
19	IMG_0789.jpg	1076	1652	1968	2021	965	1594	1838	2044	0.67657566	True Positive	281057	440687	0.64
20	IMG_0800.jpg	1009	2114	1408	2573	986	2148	1457	2631	0.54308152	True Positive	169485	241027	0.70
21	IMG_0748.jpg	1347	1195	1528	1851	none	none	none	none		False Negative	N/a	N/a	N/a
22														
23	PRECISION	0.84615385												
24	RECALL	0.64705882												
25	Average IoU	0.65												

Appendix VI: 491 Design Document

Squirrel

Design Document

Team 3

Client: Bob Thompson

Adviser: Gary Tuttle

Team Members: Abraham Contreras-Ramos, Richard Cushing, Devon Driscoll, Dan Gilbert, Cole Patton, Isaac Tegeler

Team Website: **http://sddec20-03.sd.ece.iastate.edu**

Executive Summary

Development Standards & Practices Used

- Using a hybrid agile process
  - Weekly “stand-up” meetings
  - Dynamic sprints
  - Code reviews before merging
  - Use git for version control

Summary of Requirements

- The product must scare away squirrels
- The product must not damage the house or injure the clients pet
- The product must be able to withstand bad weather conditions
- There should be some way to monitor the status of the device
- The device must be able to attach to the deck without damaging it
- The device should be relatively low in cost

Applicable Course from Iowa State University Curriculum

- Cpre 185
- Cpre 288
- Cpre 458
- Cpre 575
- EE 333
- SE 185
- LD ST 322
- Com S 227
- Com S 228
- Com S 309
- CprE 482x

New Skills/Knowledge acquired that was not taught in courses

- Developing and designing a mechanical system to move the device
- Building and testing prototypes
- Evaluation and selection of parts needed to build the device
- Working on teams with people that specialize in an area different from our own.
- Identifying and providing for the needs of a client
- Developing a test plan

Table of Contents

1 Introduction

1.1 Acknowledgement

Thanks to Bob Thompson for providing the project idea, and Gary Tuttle for technical knowledge and advice.

1.2 Problem and Project Statement

Bob Thompson has squirrels that come and chew on his deck. Bob needs to get rid of the squirrels because the damage that they are doing to the deck and house is expensive to repair. The squirrels are smart enough to ignore loud noises and Bob's dog barking at them. Therefore, we need to find some way to autonomously scare the squirrels off of Bob's deck in a nondestructive way. The device that we create must be able to withstand the outdoors and potentially poor weather conditions. It must also only target squirrels, Bob has a pet dog and it is important that the launcher does not shoot at him.

To solve Bob's problem we propose using a small turret system that targets the squirrels and shoots them with ping pong balls. The ping pong balls will be attached to a fishing line so that we can reel them back in once they have been launched. We plan to use a sensor network and a computer vision system to identify and then target the squirrels with the launcher.

1.3 Operational Environment

One of our biggest challenges for this project is making the turret able to withstand the weather. Since it will be exposed to the elements on Bob's deck we need to design it in a way that if there is a storm or bad weather the turret will not be damaged. We will also need some form of monitoring system so that if our design fails to withstand some weather we can update Bob that part of the turret has malfunctioned.

1.4 Requirements

Functional:

- The product must launch a projectile towards squirrels to scare them off the deck. The targeting system has to identify squirrels specifically so that it won't shoot at people or other animals.
- This projectile must not damage the deck, or injure the clients pet. There are windows along the house that can be easily broken and there will be traffic on the deck from other

animals and people. In case the projectile misses the squirrel, there must be zero risk that the projectile will cause harm to the property or any other animal.
- The product should automatically retrieve the projectiles that are launched
- The device should start tracking and targeting squirrels as soon as it is powered on

Non-Functional:
- The product must be able to withstand bad weather conditions. Wind should not prevent the product from rotating as normal and rain must not prevent the targeting system from identifying squirrels.
- The device should be relatively low in cost. All three devices should be produced within the base budget provided for the class.
- The device should be autonomous and require minimal manual intervention.
- The device should be simple enough that it can be reproduced and scaled to multiple devices that cover an entire deck.
- The device should operate for long periods of time without maintenance.

Engineering Constraints:
- Must be small enough to fit on deck or deck railing
- Must be possible to power in an outdoor environment
- Must be reproducible, i.e. our parts must be possible to manufacturer outside of our 3-D printed prototype
- Must be powerable by a typical wall power outlet
- Cannot have any exposed wires that could be shorted or become a dangerous fire hazard.

1.5 Intended Users and Uses

While our product is designed for Bob's situation it could potentially be used in any situation where the user wants to scare away squirrels. The only adjustments that would need to be made for each user would be the attachment that secures the launcher so that it does not move or fall over. Because our device is designed to be low-cost and reproducible it will be able to scale to any deck size.

1.6 Assumptions and Limitations

Assumptions:

- Three identical machines, each with a shooting range of around 15', will be sufficient to cover the area of the deck.
- Firing a ping pong ball toward the squirrel will be enough to deter it from damaging the deck.

Limitations:

- Machine must have a reasonable cost.
- Has to operate for long periods of time without maintenance.
- Must stay operable in strong winds and rainy conditions.

1.7 Expected End Product and Deliverables

The goal of our project is to produce a functional squirrel deterrent turret by December of 2020. Given the success of the first unit, as well as time and budget constraints; we may decide to build up to 2 more units to improve coverage of the property. The sentry device can be broken down into the following 5 subsystems: power, vision, targeting, launching, and UI.

The power system is responsible for stepping down AC power from an outlet to DC voltages that will be used to power the Jetson, motors, camera, sensors, and launcher.

The vision system is responsible for controlling the PIR sensor, additional depth sensor if needed, and the camera system in identifying the squirrels. The collection and labeling of squirrel images to train YOLO (an algorithm for detecting objects in images) is also included in this section.

The targeting system is responsible for rotating the launch box, and is used to sweep the area with the camera and line up the shot for the launching system.

The launching system is responsible for firing and recoiling the projectile. The projectile will be launched with a flywheel. The projectile will be reeled in with a motor and limit switch that winds back a fishing line attached to the ball, to pull it back to its starting position.

The UI is an additional proposed idea that would display system status over a wifi connection. The UI will be used to display error information useful for debugging or repairs, as well as information on the number of squirrels tracked over a given time. The UI is considered a bonus feature that will only be completed when all other systems are completed and will not be detailed in the rest of the design document.

2. Specifications and Analysis

2.1 Proposed Approach

Our proposed approach is to create 3 identical machines with AI capabilities to identify squirrels. Upon identification, it will shoot a ping pong ball at it to scare it away. The ping pong ball will then be reeled back using a string that will be attached. The machines will turn on when powered and begin tracking.

So far we have completed our design of the turret as well as splitting it into a collection of subsystems. The targeting system, the vision system, the reeling system, the launching system, and the power system. The functional and nonfunctional requirements are listed as test cases in 5.3 and 5.4, and at a higher level in section 1.4. The following is our proposed solution to handling each of the subsystems:

Targeting - The targeting system will consist of two stepper motors and a gearing system that is controlled by a software driver allowing programmers to interface with the targeting system by providing simple coordinate inputs

Launching - The launching system will use a flywheel to launch the ping pong ball. It will be turned off and on through a software driver.

Reeling - The reeling system will consist of a spool of fishing line and a motor. We still need to determine how to decide when to stop reeling the fishing line, but our initial idea is to use a limit switch.

Power system - We plan to use an out of the box power converter to power the motors and our controllers

Vision system - We will use a PIR sensor to detect motion, and a nvidia jetson nano with a raspberry pi to detect squirrels. If we find accuracy to be an issue in the final design we will also include a depth sensor for better targeting.

Once the PIR sensor detects motion it will trigger the camera to turn on and start an application that uses YOLO object detection to locate any squirrels it sees. Once it finds a squirrel it will send the coordinates to the targeting system which will move the fly-wheel into position. Once in position the fly-wheel will turn on. After a specified time the fly-wheel will turn off and the reeling system will turn on. Finally the reeling system will turn off once the ping pong ball has returned to the device; this control flow is shown as fig. 4 in section 2.4.

2.2 Design Analysis

So far, we've designed the basic mechanical architecture and have started ordering parts. We have just begun implementation of the targeting system and the vision system and are still working on determining what parts we will need for the other subsystems. We have also built a prototype for the fly-wheel launching system from parts we will not use in final design to verify it will launch a ping pong ball. After building the prototype for the fly-wheel we found that it was able to successfully launch the ping pong ball 15 feet without any issues. We will need to modify the launcher to better fit inside the launch box, and to be more stable.

Strengths of our current plan include:

- Minimal power usage due to cameras only being turned on if the always-on motion sensors detect movement.
- User-friendly UI to report errors and bugs

- Camera protection
- Batteries not constantly needing replacement
- Not needing to move negates navigational issues.
- Retractable ping pong ball that doesn't require extensive retrieval measures

Weaknesses of our current plan include:

- Long-term durability against environmental damages
- Power will require an ugly electric cable connecting to the house's power outlet
- Wifi connection required
- Possible tangling of fishing line on each launch and return

2.3 Development Process

We will be using an Agile development process with weekly stand-ups and three week long sprints. We have chosen an Agile process because it's flexible and gives us room to learn with each iteration and set appropriate goals/deliverables. Since we aren't very familiar with working on a project like this and working with people from different areas of study, an agile process will let us focus on small deliverables and make sure everyone has something to work on. Weekly stand ups should be sufficient to keep everyone updated; more frequent stand ups would be hard to coordinate between all six group members, but may be necessary the week of an important deadline. We also decided on three week sprints because it forces us to break the project down into smaller, manageable pieces while still giving us enough time to work on important functionality.

We have also decided against a Test Driven Development process due to the fact that a lot of the code we'll be working on is unfamiliar to us and involves a lot of configuration. This means we'll be reading through documentation and then doing a lot of experimenting. Once we have communication set up we'll add unit and integration tests.

2.4 Conceptual Sketch

The system below is a sentry device designed to identify and agitate squirrels. The primary component box (fig. 2) holds a PIR sensor, additional depth sensors if needed, power converter, the Jetson Nano, and a collection of motor controllers. The power convertor is used to step the AC voltage from a wall outlet down to the DC voltages needed to power the electrical components within the box. The sensor array is used to detect motion within the range of the sentry. The Jetson Nano is used to control all of the other devices in the system and provide object detection. The box is mounted on two clamps used to attach it to the deck. The box must be waterproof and stable to protect the electronics from the weather

Attached to the top of the control box, lies a gimbal system to control the pitch and yaw of the launcher (fig. 3) the system will be controlled using two stepper motors. After a squirrel has been detected the stepper motors will work together to move the launching system to point at it. The gimbal system will use two half geers that connect with two smaller gears on an axle controlled by a stepper motor to control the pitch. The launch box will be attached to the top of the two half gears. The yaw will be controlled with a frame that holds a larger gear that the pitch frame is mounted on, and will be turned with a smaller gear connected to a stepper motor. The gimbal system will be encased in flexible tubing that can move with the gimbal to protect it from weather.

The gimbal system will then be attached to the base of the launching system (fig. 1) which holds the targeting camera as well as the launching and retrieving mechanisms. When movement is detected by the sensory array, the Jetson will activate the camera and gimbal to sweep the area for squirrels using machine vision. The camera is attached to the launcher so that it will line up with the projectile trajectory. Once fired, a motor is used to reel back the projectile and place it into the launch position. The launch box will be weather proof to protect the interior electronics from the weather.

Finally (fig. 4) shows a block diagram of the control flow.

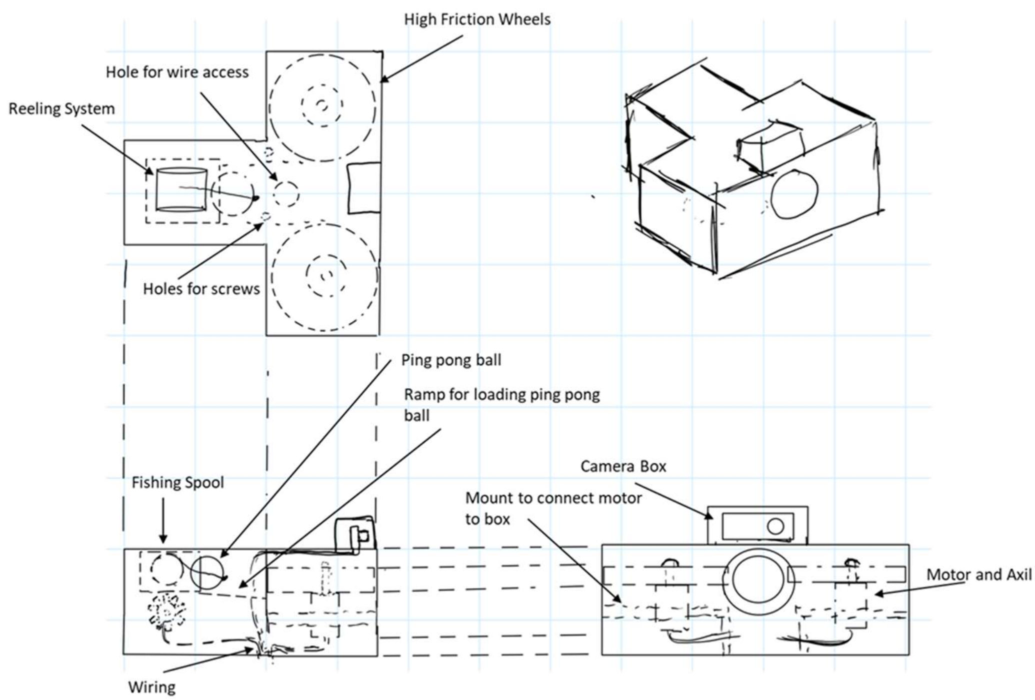


Figure 1 (above): Launching System

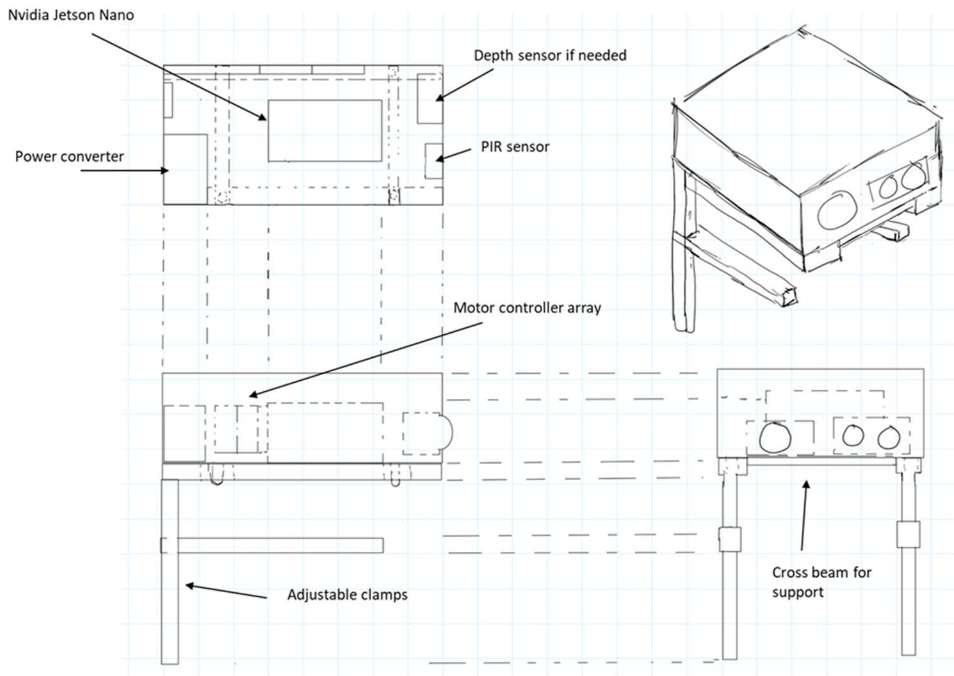


Figure 2 (above): Control Box

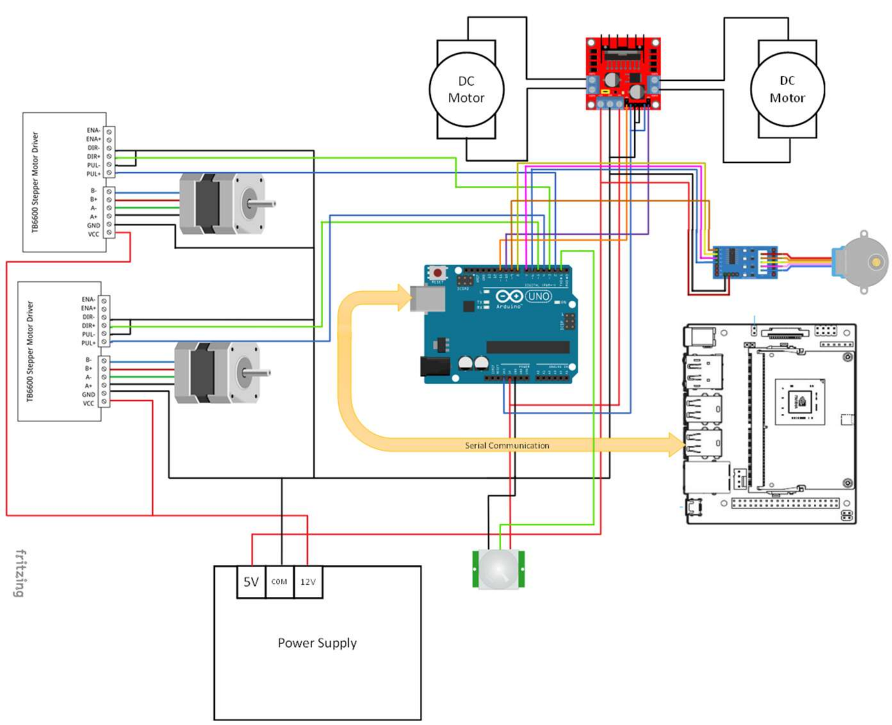


Figure 3 (above): Schematic diagram

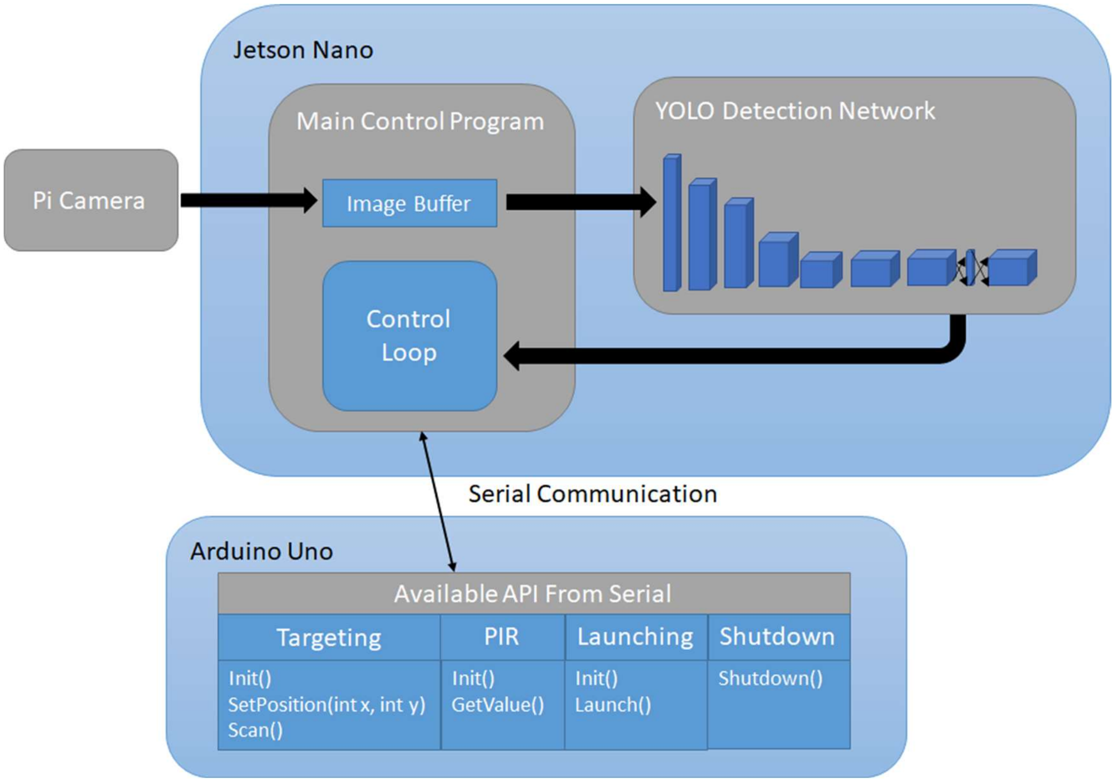


Figure 4 (above): System Architecture

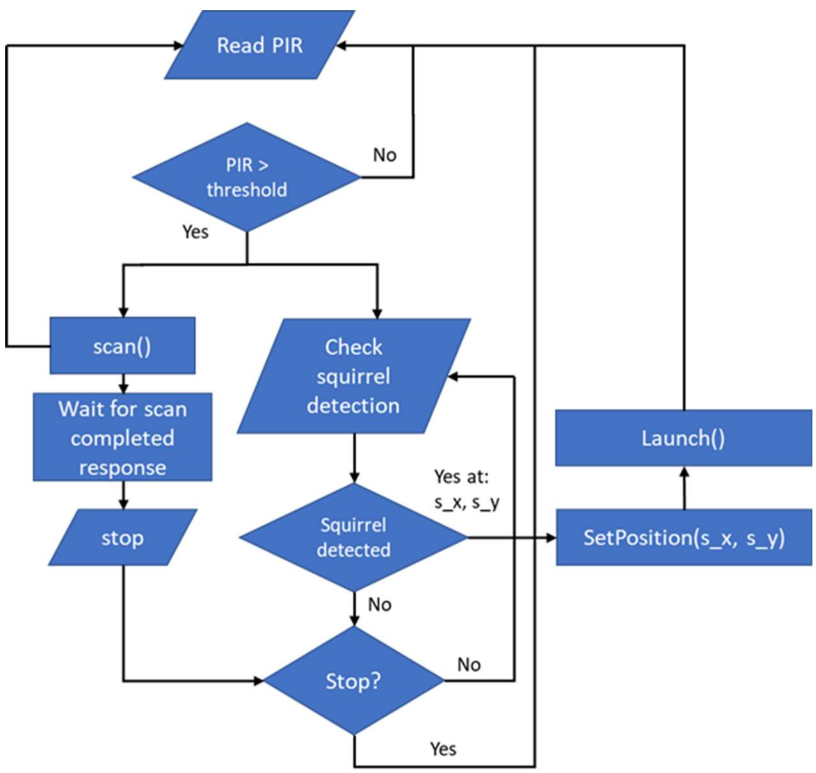


Figure 5(above): Block Diagram: Main Control Loop

3. Statement of Work

3.1 Previous Work And Literature

We intend to model the computer vision piece of our project after a similar project that was done by Peter Quinn [1]. Peter uses a product called Jevois, a camera system with a small microprocessor attached that is optimized to use deep neural networks (DNN) for object localization and classification. Peter takes the on board YOLO (you only look once) classifier and uses transfer learning to fine tune it to identify the squirrels in his garden. Transfer learning is a technique where you take a previously trained DNN and continue its training using a small data set specific to your use case. Using transfer learning Peter was able to tune the network with only a few images of squirrels in his yard and get satisfactory detection.

Using Peter's method could be advantageous to us. It allows us to offload the processing of the network to the Jevois camera, and we will not need to worry about the processing requirements for the main arduino processor that we intend to use. It also gives us a path to develop the vision network without needing to design it ourselves. While this project gives us a jump start on the vision system for our project it is not the perfect solution. The Jevois camera is very slow when using the YOLO network. An alternative that will boost our speeds for this network would be to use the Jetson nano, a full computer for embedded applications that has an onboard GPU module for DNNs. Another issue is that we will need to set up a system to collect and label images to train our network. In order to use transfer learning we will need to capture images of the squirrels on Bob's deck and off of google.

3.2 Technology Considerations

For the main controller we needed to choose between a raspberry pi, an arduino, and a Jetson nano. These controllers are very popular. This means that if we run into issues when building the sensor network and connecting the vision system there will be a lot of online resources to assist us in solving the issue. The benefit to using the raspberry pi over the arduino would be its built in wifi capabilities and that it is a full computer which makes it easier to implement the higher level computer vision pieces of the project. The Jetson nano has these same benefits but also has an onboard GPU. If we use the Jevois camera we do not need the GPU module of the Jetson and the raspberry pi would be the best option.

The computer vision system has two options. One is the Jevois camera, and the other is the Jetson nano. The Jevois camera gives us the option to decuple the image processing and detection from the main control system, but at the cost of a significant performance drop. Our research indicates that using our network on this camera would result in a frame rate of 0.5 fps while Jetson can manage around 3-4 fps, meaning it takes Jevios camera 2 seconds to find a squirrel and Jetson around .3 seconds. Both of these systems would be the largest single cost of the project with the Jetson being around $100 + $30 camera and the Jevois + pi combo being $60 + $55. So the cost is similar between the two and will not be a factor at small volumes.

The launching system has three options. A flywheel, a spring loaded gun, and pneumatics. Pneumatics are a good option because they involve no exterior moving parts for launching. However, looking online it was determined that the pneumatic system for launching would be too expensive around $200. The other two options are a flywheel and a spring loaded gun. The spring loaded gun would come fully built however it would be difficult to reload with our reel system. The fly wheel would be easier to reload, but we will have to assemble it ourselves from motors and gear systems.

We are looking at using vex kit parts for the motor systems and the gear systems as well as building the flywheel. These parts can be ordered in small volume at relatively low cost and are designed to be easy to use. One major downside is that we will have to write our own drivers since the motor modules are not designed to be used with the main controllers that we want to use. As of 04-11-2020 we decided against using the vex parts because the motor systems are very large and violate our low cost requirements. This means that we will also not be able to use the other vex parts as they are only compatible with the vex motors. Instead we plan to use a 3-d printer to build our prototype gear systems

3.3 Task Decomposition

Targeting System

1. Find or build a gearing system for turning the main axle connected to the launcher.
2. Find or build a motor system to control and turn gears.
3. Define feedback loop and develop program for turning system to specific location.

Vision System

1. Find sensors for motion detection.
2. Develop a loop to turn on the camera when motion is detected.
3. Collect and label images of squirrels on Bob's deck and from google.
4. Fine tune YOLO network.
5. Load a custom network onto the Jevios / Jetson system.
6. Feed location of detected squirrels from camera to main controller.
7. Develop controls for a targeting and launching system to fire at squirrels.

Launching System

1. Find or build a flywheel to launch ping pong balls. Modify it to connect to the targeting system.
2. Define and program control loop for launching the ping pong ball

Power System

1. Finding appropriate ADC Converter for walled power connection

Reeling System

1. Find fishing line and spool to use
2. Design release and reel system
3. Find motor and limit switch to control reeling

3.4 Possible Risks And Risk Management

Our main risk is the mechanical system used to control movement and firing of the ping pong balls. As EE/CprE/SE students we have not had any training on developing or building these systems. To mitigate this issue we plan to start development of the mechanical systems early

Another minor risk is the training and deployment of the YOLO network. From past experience we know that this is very complex and will take a long time to do. To mitigate we will also start on this portion early

Our highest risk factor currently is creating a reeling system that does not tangle easily and often when the ping-pong ball is launched. We need to think of ways to prevent tangles with things like chairs or tables on the deck. As of right now we do not have a mitigation option and thus it is deemed high risk.

Finally due to the COVID outbreak we are at risk of not completing prototypes on time. Because we are unable to be on campus it is difficult to get the parts we need, and work together on building the systems.

3.5 Project Proposed Milestones and Evaluation Criteria

Milestones

1. Create full bill of materials
2. Order all Parts
3. Finalize design document
4. Targeting system prototype
5. Launching system prototype

6. Vision system prototype
7. Power system prototype
8. Reeling system prototype
9. Train Vision system
10. Integrate launching and targeting
11. Integrate vision
12. Fully integrate system
13. Pass system test

Milestones 1 through 8 detail our goals for the first semester. Milestones 8 to 13 cover the goals for the second semester. See project timeline for more details.

3.6 Project Tracking Procedures

We will track our progress using the issues page on gitlab. We have a main (epic) issue for each subsystem, and we will create other issues (stories) that reference the subsystem issue that they belong to. The stories will track meeting notes, code implementations, and other tasks that are required.

3.7 Expected Results and Validation

This section is a high level view of our expected results. Further detail can be found as our functional and non functional tests in section 5.3 and 5.4.

Targeting System - System will turn the launcher left and right to a specified location of the programmer.

Launching and Reeling System - System will launch and reel in ping pong ball when the command is given by the programmer

Vision System - System is able to detect motion. This detection will trigger the main camera to turn on. The main camera will send object detection information back to the main controller.

Main Control - The main control system will take in the information from the sensors and send it to the UI. When the main camera sends a detection signal the main control will use the targeting and launching systems to fire in the general location of the detected object.

Power System - All systems will be left undamage through regular use.

Full System - All subsystems will synchronise to successfully detect a motion, turn on the camera, detect a squirrel, target it, launch the ping pong ball, and finally reel it in.

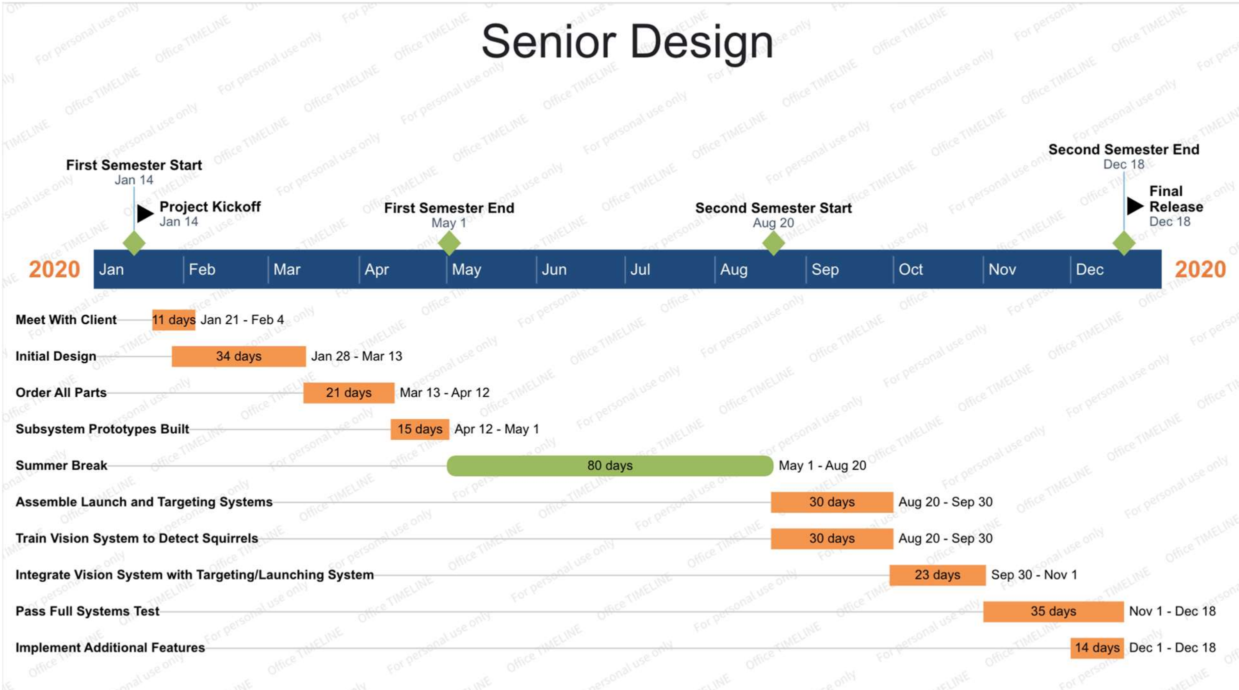
4. Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline

At the end of the spring semester we will have assembled prototypes for each of the subsystems. To meet this goal, we plan to create the full bill of materials and complete parts 3 and 4 of the design document by the end of march. In the first two weeks of April, we will order the parts necessary for the vision system and finalize the design document. In the 3rd week of april, we will finish ordering all the system parts. Over the summer we will continue to improve our subsystem prototypes and plan for the next semester.

In the second semester we will begin by spliting the group into two sections. The first section will work on integrating the launching and targeting systems, while the second section trains the vision system to detect squirrels. After the first 6 weeks of class, we will continue working together to integrate the vision system into the main project. In the final 5 weeks of the class, we will work towards passing the full system test. In the final 2 weeks we may begin testing the device on-site at the client's property.

After these tests are completed, we will implement some additional quality of life features given timing, and work on our final presentation. These features include a website for UI that can be accessed via wifi. It will report analytics on the squirrels as well as device diagnostics.



4.2 Feasibility Assessment

For our project we believe that it is feasible to complete the targeting and launching systems. We also anticipate that the motion detection piece of the computer vision system will be completed. We anticipate some difficulty in the Squirrel detection piece of the computer vision system as the technology that we are using is only four years old. This means that there are very few out of the box solutions for this piece of the system and it will take a lot of work. Furthermore, the ability to detect the squirrel and then aim towards it might take too long; if a squirrel continues moving around, actually hitting the squirrel with the ping pong ball might not be feasible. That being said, if our machine keeps squirrels from sitting down and chewing on the wood, that is still a success for our project.

We anticipate the most difficulty in reeling in the ping pong ball and preventing the fishing line from getting tangled or stuck. We expect that when there is nothing around we will be able to reel in with no issues, but if there is deck furniture we will have issues with this system.

4.3 Personnel Effort Requirements

See the Gnatt chart in section 4.1 for a timeline that shows the higher level progression of tasks and their duration. The table below will cover each task in more depth and show who in our group is assigned to it.

Start Date	Task	Description	Member/s Assigned to Task	Estimated Duration
01/21	Meet with our client.	As a group, we will schedule a time and meet with our client, Bob Thompson to discuss the project in detail and get a solid understanding of the problem.	All	1½ weeks.
01/28	Initial Design	As a group, we need to come up with an initial design for our Squirrel Turret. This involves coming up with potential solutions to detect the squirrels, target them, and then fire a ping pong ball at them.	All	1½ months
03/13	Order Parts	Once we have a solid plan, we need to order the necessary parts so we can begin building prototypes. Estimated duration includes extra time for the parts to ship to us.	All	1 month
04/12	Start Targeting Prototype	Begin assembling a mechanism to hold and aim the launching subsystem. The targeting system must account for both horizontal movement and vertical movement. The targeting system must also not impede the launcher in any way.	Isaac	1 month
04/12	Start Launching Prototype	Begin assembling a mechanism to launch and reload a ping pong ball which is attached to fishing wire. The ping pong ball must be launched at a constant velocity and should be ready to fire almost immediately after a squirrel is detected. The launching mechanism is also responsible for reeling in the ball after it was launched so it could be fired again.	Dan, Ricky	1 month

04/12	Start Vision Prototype	Begin programming the software required to detect squirrels. This involves gathering 300-1000 photos of squirrels to train a model, training a model to detect squirrels, and then setting up Yolov3 to use our custom model to detect squirrels in real time. It also requires building a driver for the PIR sensor that will activate the camera.	Cole, Isaac, Ricky	1 month
8/20	Assemble the Launching and Targeting Systems	Start assembling the launching system and the targeting system. Make sure that when assembled, the launcher can turn and tilt to all necessary angles without the line of fire being obstructed.	Isaac, Ricky, Dan	1½ month
8/20	Finish Preparing Vision System	Make sure that the Jetson Nano can detect squirrels and distinguish them from small dogs. The vision system should also be detecting objects at a sufficient FPS. Ideally 10+ fps.	Cole	1½ month
10/01	Integrate Vision System with Targeting / Launching System	Integrate the vision system with the targeting / launching system. Get the vision system to communicate with the targeting system. When a squirrel is detected, coordinates should be sent to tell the targeting system where to turn the launcher.	All	1 month
11/01	Pass Full Systems Test	Make sure everything is functioning together: detect a squirrel, turn the launcher towards it, fire the ping pong ball, and then reload the ball. Also make sure that our power supply is sufficient and fix any bugs that show up at this stage.	All	1½ month

12/01	Implement Additional Features	If time allows it, we can focus on extra functionality to improve the quality of life when using our turret. Early ideas include things like a UI for the user to see errors or be alerted when a squirrel is hit.	All	2 weeks

4.4 Other Resource Requirements

For the vision system we will need a computer capable of training the YOLO classification network, a pre-trained version of the network, and ~500 - 1000 labeled images to train it on.

The following is the list of other parts required for the turret, their anticipated costs are listed in section 4.5

- Jetson Nano
- Raspberry Pi camera
- 2 - stepper motors and controllers for targeting system
- 2 - high speed motors and controllers for fly-wheel
- Motor and controller for reeling
- Power converter for jetson and motors
- Casing and weather protection
- Gears, axles, and other mechanical
- Clamps
- Fishing line for reeling

4.5 Financial Requirements

We have a $600 budget for this project as this is how much we have allocated to us through our lab fees. However, since we need to make our design as cost effective as possible to make multiple launchers to cover the whole deck we will be aiming to keep a single launcher well under budget. The following is the expected costs for each module.

Part	Quantity	Cost
Jetson	1	$99
StepD-01	1	$14.95
Pi Camera	1	$29.33
Nano Power Supply	1	$7.50
SSD 64 GB	1	$14.99
Nema 17 Steppers	2	$27.98
DC Motors	2	$13.78
L298N H-Brige	1	$5.99
TB6600 Driver	2	$23.98
6 ' Power Cord	1	$6.16
3D-printing plastic	2	$40
Casing	1	$15
Power Supply	1	$20.77
Wirering	1	$20
Arduino	1	$15
Total		$354

5. Testing and Implementation

5.1 Interface Specifications

We will use the built in CI/CD tools in gitlab to track tests and implement unit tests that will run on each build. We have not yet determined what unit testing tools need to be used as we have yet to write any code to test it with. Our plan is to use CUnit if possible. Gitlab also seems to have the ability to run python scripts, we will use this functionality to track the performance of the vision system as detailed in the next section. For all other testing we need to do manual testing to verify that the systems work as expected. We were unable to find any automation tools that are able to do embedded systems tests where you need to verify the physical response as is required with the majority of our tests involving the targeting, launching, and reeling systems.

5.2 Hardware and software

For the machine vision, we will write scripts to test the performance of our object detection. These scripts will calculate the IoU, precision, and the recall on a validation set of labeled images that were not included in the training dataset.

Intersection over Union - The IoU is a way to measure if an object detection program's prediction is correct or not. It calculates how much of the predicted bounding box is accurate by taking the intersection and union of the predicted bounding box with the true (human labeled) bounding box.

$$IoU = (Intersection) / (Union)$$

It starts with a threshold, usually around 50%. If the IoU is more than 50%, then it's considered a "True Positive", or it correctly detected a squirrel. If the IoU is less than 50%, then it's considered a "False Positive", or it detected a squirrel but there is no squirrel there.

Precision - The precision of an object detection program is a measure of how many "detections" were actually correct.

Precision = (True Positives) / (True Positives + False Positives)

Or

Precision = (True Positives) / (All Detected Positives)

For example, if you had a model with 10 squirrels, and it detected 8, but 3 of those detections were wrong, the precision value would be ⅝ or 62.5%.

Recall - The recall of an object detection program is a measure of how many items it correctly detected.

Recall = (True Positives) / (True Positive + False Negatives)

Or

Recall = (True Positives) / (Number of Positives That Should've Been Detected)

For example, if you had a model with 10 squirrels, and it detected 8, but 3 of those detections were wrong, the recall value would be 5/10 or 50%.

All the measurements will be used to help test the machine vision software to determine when it's accurate enough to be integrated with the targeting / launching systems.

5.3 Functional Testing

The following are the hardware and software functional requirements for each subsystem as well as what testing strategy we plan to use to verify that the requirement is met

Targeting

<u>Software</u>

Unit Test (1) - Driver will send signals to motors that control pitch and yaw based on a degree input.

Unit Test (2) - Driver will return a message when the position has been set.

Integration Test (3) - Driver will accept coordinates from the vision system and set pitch and yaw based on the coordinates.

System Test (4) - When a squirrel image is shown to the camera the driver sets the pitch and yaw to appropriate angles to fire at the squirrel.

Hardware

Acceptance Test (1) - The motors will move the mount for the launching system to the correct position when it is set in the software.

System Test (2) - When a Squirrel image is shown to the camera the motors move to the correct position to fire at the squirrel.

Launching

Software

Unit Test (1) - Driver will turn on the fly wheel motors to the correct speed and direction when the launch command is sent.

Unit Test (2) - Driver will turn off the motors after a specified time interval in which the ping pong ball should have been launched.

Unit Test (3) - Driver will return a message when it has finished launching the ping pong ball and the flywheel has shut off.

Integration Testing (4) - When the targeting system has finished positioning, the launching system will fire.

System Test (5) - When a squirrel image is shown to the camera the ping pong ball will be launched after the targeting system has finished positioning.

Hardware

Acceptance Test (1) - The ping pong ball is launched when a command is sent in the software.

Integration Test (2) - When the command is set in the software to launch both flywheel motors turn on in the correct direction.

System Test (3) - When coordinates are set from the vision system the launching system and targeting system move and fire at the coordinates specified.

Reeling

Software

Unit Test (1) - The driver will set the motor to the correct direction and speed when the reel command is sent.

Unit Test (2) - When the driver receives information that the ping pong ball is stuck it will turn off the motor and return the stuck message.

Unit Test (3) - When the driver receives information that the ping pong ball is entirely reeled in it will turn off the motor and return the success message.

Unit Test (4) - After a failed launch of the ping pong ball (a possible jam), the reeling system will attempt to reel the ball in and if the ball is stuck it will stop the motor and return a jammed message.

Unit Test (5) - If the ball is launched and then broken on return, the motors will turn off and not allow the broken ball back into the machine (broken bits of the ball could build up in the machine and cause issues) while returning a broken ball message.
Integration Testing (6) - After the launching system has fired the reeling system will attempt to reel in the ping pong ball.

Hardware
Acceptance Test (1) - When the software signal is sent to reel in the ping pong ball the ball is reeled in.
Integration Test (2) - When the software signal is sent to reel in the ping pong ball the motor to reel in the ball turns on.
Unit Test (3) - When the ball has returned the reeling system sends a signal to the software
Unit Test (4) - When the ball has returned the reeling system turns off its motor.

Vison

Software:
Unit Test (1) - The motion detection driver will send a signal to turn on the camera when it receives data that motion has occured.
Unit Test (2) - The camera turns on when it receives the signal that motion has been detected
Unit Test (3) - The camera turns off after a squirrel has been seen or a specified time period has passed.
Unit Test (4) - The correct coordinates are returned from a squirrel image.
Integration Testing (5) - The targeting system moves to the location of the squirrel in the image
Acceptance Testing (6) - When a squirrel walks by the vision system detects the squirrel and the system fires a ping pong ball at it.

Hardware:
Acceptance Testing (1) - The camera feed is turned on when motion is detected, and turns off after a specified amount of time or a squirrel is detected.
Integration Testing (2) - The PIR sensor continuously sends data to the Jetson and it is received by the driver.

Power

Hardware
Acceptance Testing (1) - All motors and sensors receive correct power allocation.

Integration Testing (2) - Checking voltages on each node being connected

5.4 Non-Functional Testing

Unit Test (1) - After a target (a squirrel) is acquired, the targeting system lines up within 1 second so the ball can be launched before the squirrel moves away.

Unit Test (2) - During low visibility conditions (light fog or rain) test the targeting system to see if it can still effectively find squirrels.

Unit Test (3) - Test the whole system in cold or very hot conditions to see how well each system performs.

Unit Test (4) - See how the vision and targeting systems function when there is a glare from the sun or during imperfect lighting conditions.

Acceptance Test (5) - Test that system is water resistant. Spray water on the casings when empty and see if any water gets inside.

Acceptance Test (6) - Vision system reaches above 50% IoU score, and maintains a precision and recall score above 80%.

Acceptance Test (7) - System should run without crashing for a long duration. Run for 8 hours without getting stuck or breaking.

Acceptance Test (8) - System detects and targets squirrels without human interaction.

5.5 Process

We have built a prototype of the flywheels to test the drag on the ping pong ball, how it would affect the distance we are able to launch it, and which way is best for loading the ball for launch. To mitigate the possibility of the string getting caught in the flywheels. We found the drag of the string did have a very slight impact on the distance, but it was tolerable in less than 12 inches. Loading for launch, we found the ball being reeled down from above seemed the most promising as the string stays away from rotating servos of the wheels.

5.6 Results

We found that flywheels were a valid launch mechanism and it was possible to launch without getting the string wrapped around the wheels.

Machine vision test results:
The squirrel detector was testing on a set of 19 images. Three of the images did not contain a squirrel, one image contained two different squirrels, and the remaining 16 images each contained a single squirrel, each in different locations and positions.

> **Recall -** Percentage of total squirrels detected. The squirrel detector correctly identified 11 out of 17 squirrels, giving a recall of 64.7%. The recall was originally closer to 80% when using a significantly larger detection model, but the larger model was causing the detection to run very

slow. By switching to a smaller model we lost 15% recall, but got a necessary 400% increase in the fps of the detection.

Precision - Percentage of correct positive results. The squirrel detector correctly identified 11 squirrels and incorrectly identified 2, giving a precision of 84.6%, which is very good. A concern from this calculation is that the test set only contained three images without squirrels, and two of those three images came back with a false positive. This could imply that adding more images without squirrels to the test set would give us more false positives, and therefore a lower precision. However, both of the false positives had a confidence rating that is well below 50%. The average confidence rating of the 11 true positives was 67.2%, so by increasing the minimum confidence rating for a squirrel to be detected to around 50%, we will greatly reduce the number of false positives without losing any of our correct squirrel detections.

Intersection over Union - The percent of overlap of the bounding boxes between where I manually detected a squirrel and where the computer detected a squirrel. The average IoU for detected squirrels was 65%. The cutoff for a true positive is 50% so, the 65% is more than high enough to ensure our detection is correctly identifying the squirrel's location. The IoU has room for improvement, but in every tested case except one, the detected bounding box is in the same location as the manually inputted one, except it's bigger (the edges from the box are uniformly further away from the center). Since we're only looking for the center of the bounding box, we are still getting very accurate locations to send to the targeting systems.

Detection Speed - The final detection algorithm runs at just over 5 fps. There is a relatively long startup time that consists of 8.5 seconds to load the detection model and it's weights, and then 20 seconds to start up the camera and start running the detection. The load up time, while taking longer than what was hoped for, isn't a large concern since it only needs to be started once, and then can run indefinitely. Once started, each frame from the camera is captured and run through the detection algorithm in .16 to .25 seconds.

6. Closing Material

6.1 Conclusion

So far we have completed our design. We have begun working on prototypes for the launching system, and have ordered parts to build prototypes for the vision and targeting systems. We need to find a solution to help Bob keep squirrels off his deck, and to do this we will build a turret system using the Jetson nano to find squirrels, shoot a ping pong ball at them with a flywheel, and reel it in.

6.2 References

[1] P. Quinn, *Squirrel Deterrent*, Hackaday.io, April 28, 2018. Accessed on: February 23, 2020, [Online]. Available: https://hackaday.io/project/156926-squirrel-deterrent/details