Date: Fri 10/5/18	**Location**: Belbas House	**Purpose**: Team Meeting #8

Attendees: Luke, Mark, Zach, Joel, Mariah, Ben, Andrew, Mr. Lee, Coach Belbas

Agenda:
- Team shirts
- Funding
- Notebook updating
- Design Meeting for Drive Modules

Reflections:
We did a design review during the meeting for the robot's chassis. Below are the notes we discussed.

Notes for chassis design:
- The mechanical team presented the drive module to the rest of the team
  - There are two versions of the module. One version for opposite corners.
  - The plan is for the drive modules to slide onto two protruding prongs of an H-shaped chassis, similar to last year's design
- Strengths
  - Modularity
  - Allows for a simple chassis which is easy to design and manufacture. Made from HDPE. It would use a similar "stationary fork 'H' concept" from last year.
  - Easily 3D printable
- Weakness
  - Too many pieces to assemble
  - Cutting the carbon fiber pieces, drilling them, getting them lined up would be difficult.
  - We cannot add an additional motor support because we don't want the motor taking any load to the motor gearbox connection.
  - There are weak points with the back channel on the drive module where it attaches to the chassis.
  - The motor hits the sprocket box. Raise the motor.
  - Need support between the two sprocket box sides -- mechanical knows about and has a solution
- Opportunities
  - We want to make a whole assembly with three main parts so the idea is to go with one piece for each main part. It will be stronger and possibly quicker.
  - The sprocket box would form a parallelogram shape
  - Good candidate for Fusion 360 generative design to produce optimal design with minimal mass and maximum strength
  - Thicken the sprocket box around the axle to act like a spacer to keep the wheel from sliding back and forth -- alternatively, thicken the wheel-conversion pieces in similar fashion.
  - Extend the front pieces of the sprocket box and place a piece of carbon fiber in between to connect the two sides and provide support.
  - Replace the single brace on the back of the sprocket box with two stubs and connect the two with a sleeve of carbon fiber
  - Mirror sprocket box shape on other side of wheel to simplify design and to provide better anchoring points for cross bracing between the sides of the drive module -- an issue with this modification is that it sticks out into the inside where mechanisms might be.

- Joel's plan is to have the motor mount extend out from the chassis
  - Have a piece that extends from the drive module meet a similar piece extending from the chassis. The two pieces would be connected by a carbon fiber sleeve.
  - Joel subsequently said it was a bad idea because there would be too much force on the sleeve is a weak point and could easily break. His idea would also make the chassis longer.

- Mark's suggestion to Joel's idea is to
  - Since the connection point is most likely to break in the vertical plane, we can thicken the walls above and below where the chassis prongs insert. This idea most likely eliminates the need for a carbon fiber sleeve.

- Chassis Shape:
  - An H frame with four drive modules (one at each corner)
  - Use HDPE sheets
  - Have the cut-in parts of the H be as shallow as possible in order to have as big an area for attachments as possible.

- Naming Conventions/Part Organization:
  - Issue:
    - Names are too long
    - Can't distinguish between parts from the name of the part
    - Too many folders
  - Solution:
    - Organize parts in folders --
- folders are organized by system and subsystem (chassis, mechanism X, etc)
- Inside a subsystem, have folders for each major iteration of the part (gen 0, gen 1, etc)
- Inside each generation folder, have three folders: Working, Released, Not Used
    - Name parts according to a standard naming nomenclature
- Date - short, specific name (10/5/18 Wheel Conversion)
- Use spaces and capital letters in part names
- Notes for Coach:
  - Chassis Team:
    - Lead: Mark
    - Support: Joel, Ben, Zach, Mariah
  - Attachments:
    - Lead: Zach
    - Support: Mark, Joel, Ben
  - Wiring:
    - Lead: Mariah
    - Support: Whoever is programming at the time :)
- Assignments for today:
  - Drive Modules (Zach and Mark): After the design analysis we had a good idea of where the drive modules need to be so we got right on that. The big change in our assembly was the reducing of parts into single part pieces. Our assemblies before contain many parts that we would have to manufacture ourselves and this would be a problem with cutting and drilling each part the same as the last. So the assemblies will now just contain three parts that we will 3d print insuring the placement of holes and dimensions. Also, this will make the whole assembly easier to replicate from corner to corner in the full chassis assembly.

- Chassis Base Plate: (Joel and Mariah) Today was very influential day in the progress of our robot. We began our Gen 0 chassis assembly. We began by importing last year's (2017-18) chassis into a new assembly, and then worked from there. However, as with all CAD assemblies, there were setbacks. An issue with the constraints in the motor/wheel assembly caused the entire unit to not act as one piece. Instead, individual parts would detach from their parent assembly whenever constraints were applied. We were able to work around this by constraining the chassis to the mounts instead of *vice versa.*

  The next version of the motor/wheel assembly will be made from much fewer individual parts which will allow for a more manageable amount of constraints. Mariah was also able to continue practicing her CAD skills today by learning more about assemblies. Regardless, this is the current state of our Gen 0 chassis assembly!
- Vex Conversion Pieces: Luke and Ben
  - Conversion Piece Subgroup (Luke and Ben): While Mark and Zach were presenting their sprocket box, Mr. Lee pointed out that instead of going through the trouble of adding spacers, to just make that part of the wheel conversion pieces. Above are the images of the parts that Ben adjusted today.
  - Drive Modules (Zach and Mark): After the design analysis we had a good idea of where the drive modules need to be so we got right on that. The big change in our assembly was the reducing of parts into single part pieces. Our assemblies before contain many parts that we would have to manufacture ourselves and this would be a problem with cutting and drilling each part the same as the last. So the assemblies will now just contain three parts that we will 3d print insuring the placement of holes and dimensions. Also, this will make the whole assembly easier to replicate from corner to corner in the full chassis assembly.

Date: 10/6/18	***Location***: Lee House	***Purpose***: Programming Subgroup

Attendees: Andrew, Mr. Lee

Agenda:
- Work on fabricating a software skeleton

Reflections:

We first planned out which classes we would need based on what the robot would be composed of (the different subsystems). So for example, there is a chassis class which has two subclasses: mecanumChassis and tankChassis. We also added several new classes that represent major mechanisms or functions that we know the robot will have to have. The added function classes are:

- FieldVision -- this class will handle the vuforia and openCV tasks, such as finding the gold minerals and navigating on the field using the images on the field walls
- MarkDeploy -- this class is for the mechanism that will deploy the team marker into the alliance depo
- MineralProcessing -- this class will handle the mechanism that will suck in minerals and deposit them in the lander

We also added a blank autonomous class that uses a mecanum chassis.

Our code structure:

We have a robot class called RuckusBot. RuckusBot has a chassis class which can choose one of two child classes: TankChassis or MecanumChassis. Our autonomous will be using the MecanumChassis. There are also several classes that represent functions that are talked about above. See our software documentation section for detailed class diagrams.

Date: 10/7/18	**Location**: Tate House	**Purpose**: Programming Subgroup

Attendees: Ben

Agenda:
- Programming a basic Mecanum Teleop

Reflections:

I attached 4 mecanum wheels on TestBot to start creating a 4 wheel mecanum Teleop for the upcoming Maker Fair and started working on a basic teleop . The robot phones would not connect to the expansion hub:

I tested the same phones, wires and battery on my robot and found that it connected successfully. TestBot might have a bad expansion hub so attached the mecanum wheels on my robot instead in order to program the teleop. I have two separate programs, one for a 4 wheel drive and another for strafing. I will be combining both into one class and then adding a hardware-map class. From there I will see what other areas I can change / enhance.

This section is intentionally left blank.

Date: Mon, 10/8/18	**Location**: Belbas House	**Purpose**: Team Meeting #9

Attendees: Mark, Zach, Andrew, Ben, Mariah, Mr. Lee, Mr. Bell, Mr. Belbas, Coach Belbas

Agenda:

- Deep Dive on the Collector and Arm Attachments

Reflections:

Today we did a Deep Dive brainstorming session on the collector and arm attachments for our robot. We each brainstormed, sketched, and presented our ideas to the team. We ranked each idea by voting how well it met the following criteria: speed, accuracy, simplicity to build, ease of operation, weight, and durability. At the conclusion of the meeting, we voted that it would be best for our team to expand upon the idea of a rubber band intake attached to an arm/linkage that can protrude out to collect minerals and then can rotate over the robot to deposit the minerals into the cargo hold. This will prevent us from having to waste time turning the robot around each time we deposit a pair of minerals.
Our various Deep Dive sketches are detailed below:

Andrew's Designs:
Design 1 is two things:

- The first is several proposed ways to mount the REV hubs. On is to lay them flat side by side on a raised platform. The second idea was to place them in a tower one on top of the other. The hubs would be on rails so they could slide out for easy access.
- The second design idea on the page is to place a camera running vuforia on a swivel mount on the side of the robot to give the camera roughly 180 degrees of vision. This would help with locating the navigation images on the field wall.

Design 2 is about the grabber that will grab the lander for when the robot hangs. I suggest the gripper part be inset. I noticed that when other teams' hanging arms latched onto the lander, the robot tilted inward until the arm rested against the lander. My think we should design the arm to be already resting against the lander. That way the robot won't tilt when it is hanging.

(*To see the actual sketches of these designs, refer to the following inserted pages).*

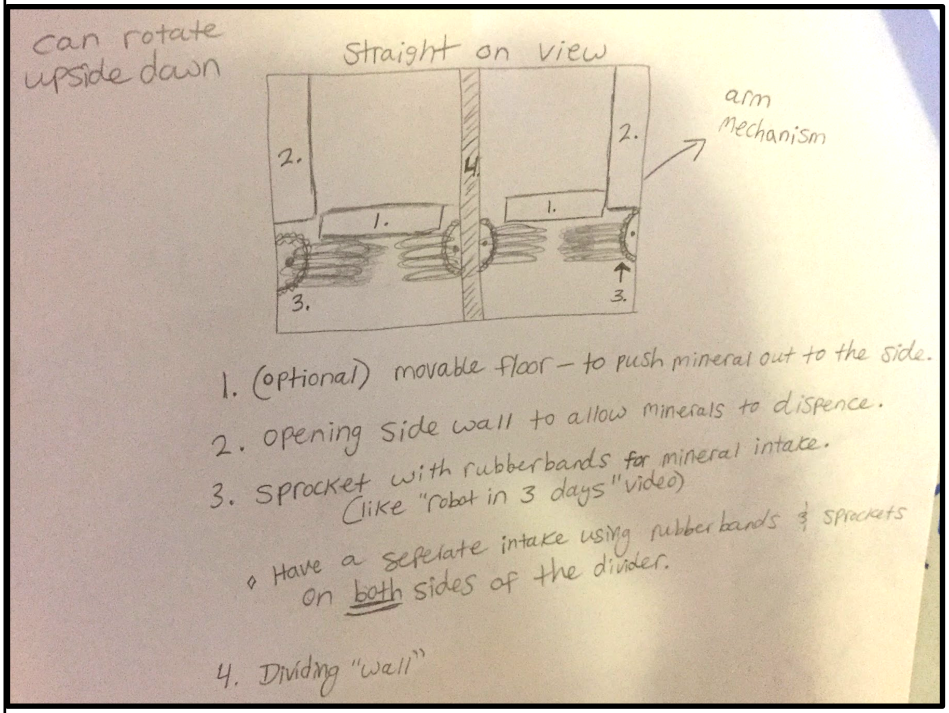
Design 3 is just several different concepts of we could deliver the minerals from ground level to the lander. Two ideas are variations of an arm and the third is a ramp that leads to an elevator with a dumping mechanism.

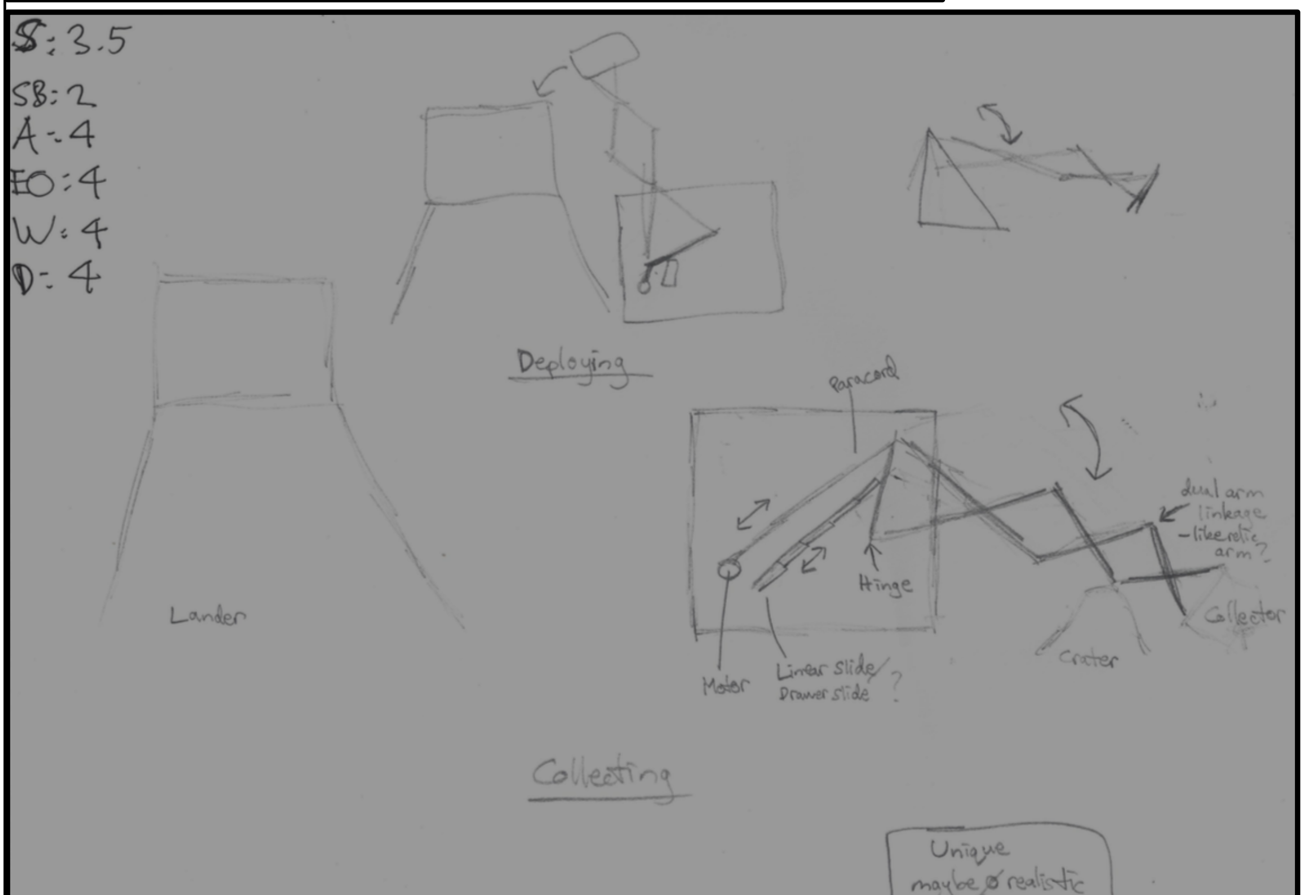
Design 4 is a mineral intake system that only allows cubes via a purely mechanical sorting system. While brainstorming with engineers at Jacobs, we discovered that the cube minerals will straighten no matter what angle they come in at with the above intake system. The two prongs straighten out the cubes so that they come in perfectly straight.

Zach's Designs: This design consists of a carbon fiber dual linkage arm (like our relic arm from last year). A linear slide/drawer slides would push the linkage forward on its rotating hinge, and paracord coiled around a motor would pull it back to then deposit the minerals into the cargo holds. It's probably not the most realistic because of its complexity to build, but it is unique.

Mariah’s Designs:

This design uses sprockets and rubber bands to take in the minerals. A divider in the middle separates the two pairs of rubber band mechanisms. This allows for quicker intake and possibly allows for picking up two minerals at a time. As an option, the walls and/or floor of the grabber could open up and dispense the minerals as needed.





Date: Fri, 10/12/18	**Location**: Belbas House	**Purpose**: Team Meeting #10

Attendees: Mark, Zach, Andrew, Ben, Mariah, Joel, Luke, Coach Belbas, Mrs. Tate

Agenda:
- Prepare for Maker Faire
- Prepare Testbots for Maker Faire
- Skype with RoboSapiens
- Brainstorm mechanism for collecting/deploying

Reflections:

Drive Module (Mark): Over the last week I worked on refining the drive module design from many parts down to three manufactured parts. I also provided more supports going from one side to the other. The big problems with the last design was how many small carbon fiber pieces we would need to cut and drill. The new design fixes that by 3D printing the whole piece and then attaching each side bytwo carbon fiber channels.

Before: After:

Maker Faire Prep (Zach): Today I updated and printed out flyers for our team and for ASSiST to hand out at the Maker Faire tomorrow. We will hand them out to people interested in FIRST as well as to companies for potential sponsorship for both our team and for ASSiST.

Later, we skyped with the RoboSapiens, a rookie team, to help them with some mechanical problems they were experiencing. We showed them several other options for assembling their axles besides using set screws, which they currently use and are having problems with. Their set screws are slipping and gouging their axles, so we helped them brainstorm different solutions to fix that problem.

Programming (Andrew, Ben, Mrs. Tate): We spent the meeting troubleshooting a bricked REV Expansion Hub and doing final setup for the two testbots in preparation for Maker Faire. The REV hub got bricked when a firmware update failed mid-installation.
Ben called REV and they were able to walk him through fixing the REV Hub.

Maker Faire Prep (Mariah): Today, I made sure all of our batteries and phones were charged and ready for Maker Faire tomorrow. I also added mecanum wheels to one of our testbots, because it did not have any wheels, due to a previous test.

Mechanical Attachments (Mark, Zach, Mrs. Belbas): After the meeting, we further brainstormed on the design for our collection and deployment mechanism. Previously, we had abandoned the idea of an articulating arm due to the hardships we experienced several years ago with it. However, we realized that the reason for the struggles was because the three degrees of motion were each programmed to be manually controlled, requiring near perfect driving skills and excessive adjusting to get the arm to a desired position. To fix this, we would need a fluid program that could move the arm to a desired location using trigonometry instead of merely fine tweaking each servo. We figured that this would be possible since the programmers' skills are at a higher level now. Having an articulating arm would allow for agility when collecting from the crater, which will probably be useful when other robots are trying to collect at the same time as us. Furthermore, the arm could fold up and we would be able to drive under the lander. This might come in handy if there is a lot of robot traffic pinning us to the lander or if we wanted to collect from the other crater. Instead of going around the lander, we could go under it, thus saving time.

Date: Mon, 10/15/18	**Location**: Belbas House	**Purpose**: Team Meeting #11

Attendees: Mark, Zach, Andrew, Ben, Mariah, Mr. Lee, Mr. Belbas, Coach Belbas

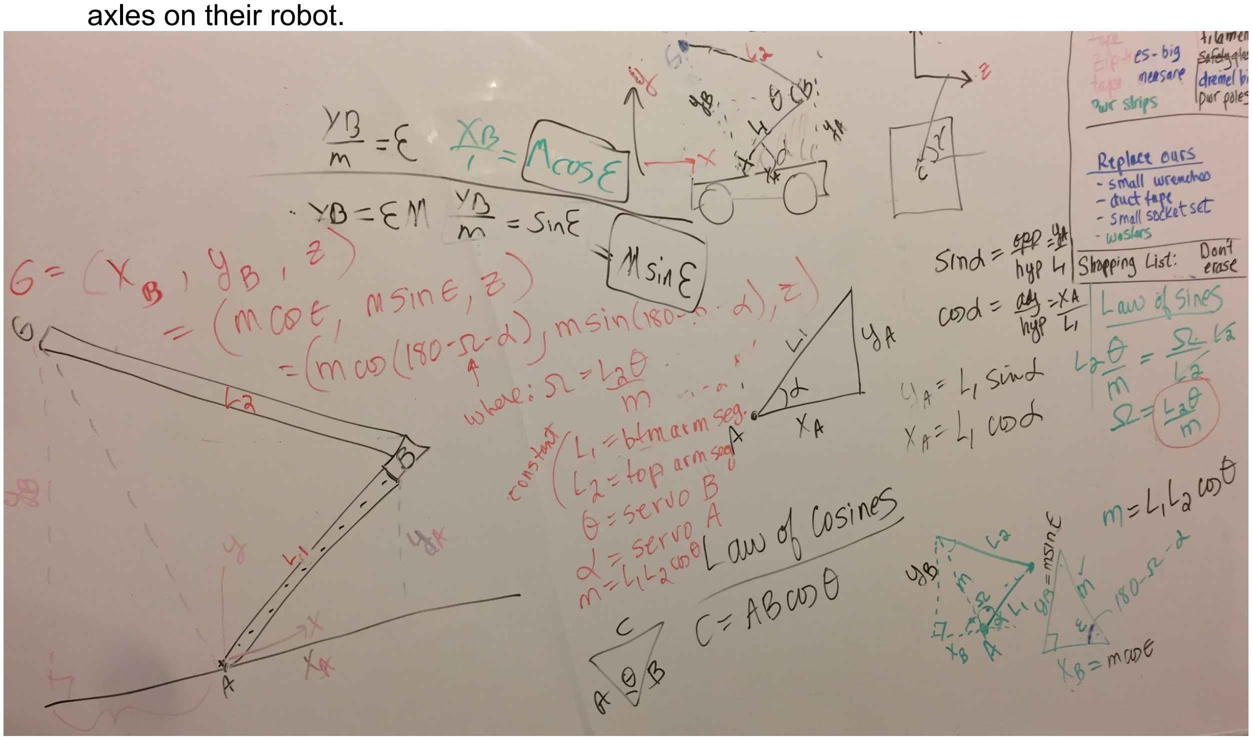
Agenda:
- Work out Trig for Articulated Arm
- Help Robosapiens with Mechanical Issues
- Work on Finishing Prototype of Articulated Arm

Reflections:

Programming (Andrew, Ben, Mariah): Except for the thirty minute pre-meeting stand up talk, we spent the meeting working through how the articulated arm is going to be controlled. First we addressed how the arm would move and what joysticks and or buttons on the gamepad would be used. We boiled the controls down to two options that each use two joysticks. Option A is to have the left joystick control side to side and forward and backwards motion (x- and z-axis) and have the left joystick control the up and down motion (y-axis). Option B is to have the left joystick control up and down and forward and backwards (x- and y-axis) and have the right joystick control the side to side motion (z-axis).

Our goal is to have the motion of the arm be just like a grabber game. Move the joystick diagonal and the arm moves diagonal while still preserving its current y position. Or move the arm up or down while still preserving the (x,z) position. To accomplish this smooth motion requires a lot of math and that's what we spent the majority of our time on: trigonometry with Coach Belbas. We weren't able to finish before the meeting ended, so we'll have to pick up next meeting where we left off.

- Mechanical (Mark, Zach): Today Mark and Zach -
  - Finished building the articulating arm out of channels from various kits. This is the prototype stage and will give the programmers something to work with.
  - Brainstormed with the programmers how to program the motion of our arm.
  - Helped the coach from Robo Sapiens brainstorm different solutions to several obstacles his team was having. The main issue we helped him through was that the set screws were slipping and gouging the axles on their robot.



Date: Tue, 10/16/18 - Wed, 10/17/18	***Location***: Lee House	***Purpose***: Programming Subgroup

Attendees: Andrew, Mr. Lee

Agenda:
- Incorporate Ben's Code into Current Class Structure

Reflections:

Andrew: Ben had written a mecanum teleop class that we used to run the mecanum testbot at the Houston Maker Faire this past weekend. Because he was working with a class structure different from the one we're going to ultimately use, I needed to go in and take his code and incorporate it into the class architecture that we're going to be using. While looking at Ben's code, dad and I found one minor bug and also found that Ben had used a different kind of mecanum code than we usually use. His code used the basic two-joystick tank drive system with a strafing ability pasted in. Our drivers are used to a different mecanum driving system with the joysticks, so I needed to adapt the code..

I ended up inserting last year's mecanum calculations into the code, which allows for omnidirectional movement and spinning at the same time. I also implemented a piece of code that fixes a semi-bug in last year's mecanum code. Because the motors can't take power values greater than one, and the joysticks on the gamepads do give power values greater than one, the power values to motors have to be somehow reduced to an acceptable magnitude. Previously, we just clipped any power values greater than one. The problem with this is that you only reduce the power commands that are greater than one. So when some values are reduced, but others aren't, your final movement vector changes. The new code (sourced from: http://www.chiefdelphi.com/media/papers/download/2906) checks to see if any power values from the joysticks are greater than one. If so, the code takes the highest value and divides all the power values by the highest command. This takes the greater than one power command down to one and also reduces the other power values by the same fraction, preserving the ratio of the power values while bringing them down to an acceptable magnitude.

Lastly, I inserted some basic drive code into the drive methods for autonomous. All of these drive methods gather the motor direction from the imputed power (ie, positive power, the robot drives forward; negative power, the robot drive backwards). I added three drive methods: drive (forward and back), strafe (left and right), and pointTurn (spin left and spin right).

Date: Fri, 10/19/18	**Location**: Belbas	**Purpose**: Team Meeting #12

Attendees: Mark, Andrew, Ben, Mr. Lee, Joel, Luke, Coach Belbas

Agenda:

- Assemble Motor/Wheel Sections
- Assemble Chain for Drive System
- Programming Arm

Reflections:

Here are the values that our articulated arm will use in degrees for programming.

	Swivel	Elbow	Shoulder
Max Front Reach:	0 deg.	180 deg. = .75	0 deg. = .08
Max Vert. Reach	0 deg.	180 deg. = .75	90 deg. .18
Sleep Pos.	0 deg.	0 deg.	180 deg.
Back Grab	0 deg.	180 deg. = .75	180 deg.

Joel: While Mark, Zach, and Luke were at Jacobs using the router to cut the chassis, I began general mechanical work. First, the arm prototype was attached to a solid piece of plywood for ease of use for the programmers. Then I began assembling the other drive modules using our newly 3D printed pieces of translucent blue PLA from Polyprinter. When they arrived back from Jacobs, Luke and I worked on different mechanical priorities. We discovered that the drive modules did not slide onto the chassis per the plan, so Luke began filing the chassis to allow for the drive modules to move freely. He took the chassis home and will continue working on it from home. Using callipers, we found the following issue: the chassis was supposed to have the following dimensions, 0.5x0.5", but was actually 0.475x0.51". This infinitesimal incongruence caused cataclysmic issues. The chassis will return on Monday.
During this time, I worked on the chain and tried to find the equilibrium between friction and tightness. I found that the distance between the sprockets from which the chain will hang was such that I could not find a perfect amount of links.
There are two possible solutions to this issue:

1. We could modify the CAD and move one of the cogs a small distance further or nearer to the other.
2. A tensioner could be added

When working with chains, it is imperative that the distance between the cogs is measured.They are very different from belts in that sense.The chains will return Monday.

This section is intentionally left blank.

Date: Sat, 10/20/18	**Location**: Belbas	**Purpose**: Programming Subgroup

Attendees: Andrew

Agenda:
- Investigate Pixy2

Reflections:
I bought a personal Pixy2 camera to play with (made by CMUcam). I figured out that the Pixy2 will not recognize white objects (ie: the white minerals) because it uses white to set the white balance. Makes sense. The Pixy2 can, however, see the yellow/gold color that the gold minerals are made out of. I also noticed that the Pixy2 can see a lot better and doesn't pick up as many bogus readings when it uses its built-in LEDs to illuminate the target area. These LEDs are, according to the FTC rules, legal. This means that the Pixy2 appears to be a viable option for detecting and locating the gold mineral during autonomous.

The first picture (below on left) is what Pixy2 sees without the built-in LEDs.
The second (below on right) is what Pixy2 sees with the built-in LEDs turned on.

I also looked at FTC's instructions on how to set up an I2C Driver and at CMUcam's I2C code; I then began drawing up an I2C Driver class in our code. I'm still learning it all, but integrating the Pixy2 into our code as an I2C device is possible.

Date: Sun, 10/21/18	**Location**: Plew House	**Purpose**: Mechanical Subgroup

Attendees: Joel

Agenda:
- Prepare four chain sections for the robot drive system.

Reflections: Joel: The optimal length was found and the four chains are ready, but, there were complications. Our current drive system (pictured right) will not allow for a perfect chain length. The pictured length may result in slipping; however, when the chain length is decreased by a single link, it becomes too small to encompass the entire distance. As discovered on Monday, there are two possible solutions: either increase or decrease the distance between sprockets, or add a tensioner.
Chain can be a difficulty to work with.

This section is intentionally left blank.

Date: Mon, 10/22/18	**Location**: Belbas House	**Purpose**: Team Meeting #13

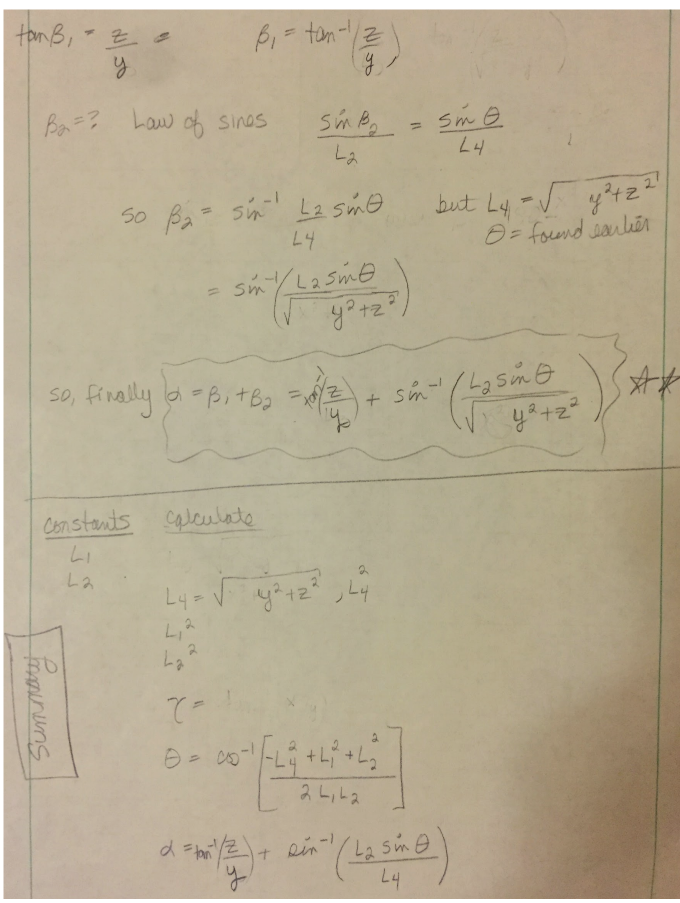
Attendees: Andrew, Ben, Mariah, Mark, Zach, Coach Belbas, Mr. Bell, Mr. Lee

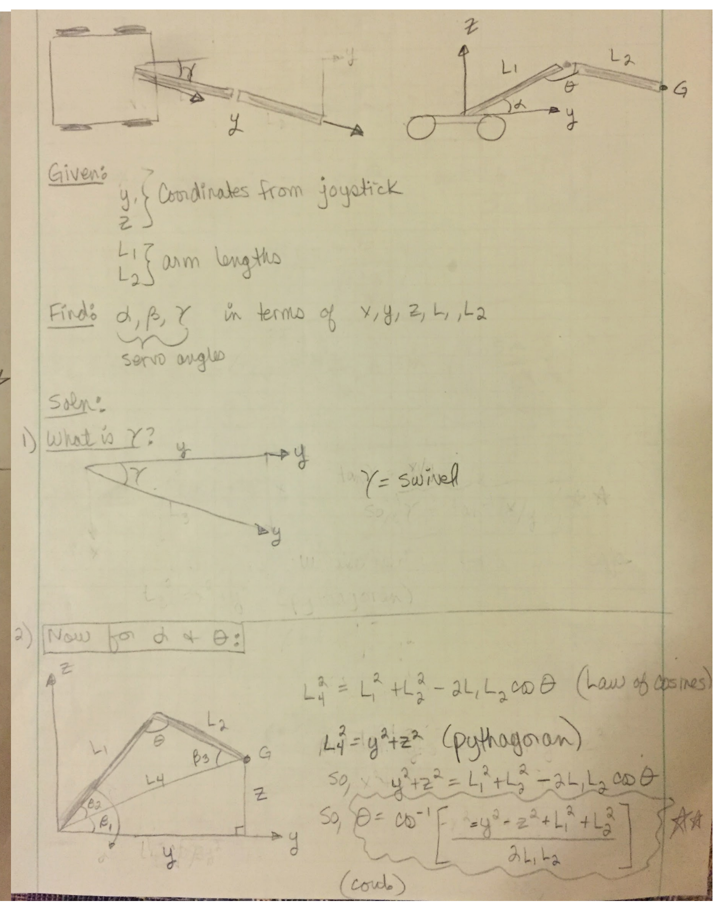
Agenda:

- Complete Chassis Assembly
- Go Over Trig Calcs
- Discuss Code Artitecture

Reflections:

Programming (Andrew, Ben, Mariah, Mr. Lee): We started by going over the class structure that Andrew created for the team and making sure everyone knew how it worked. Then we moved on to looking through the trig calculations from several meetings ago and how we were going to implement them in the code. While doing that we discovered that the x value in the calculations wasn't needed. Taking the x out simplifies things a little bit more. For starters, we decided to get one servo moving, so Ben worked on that while Andrew and Mariah grabbed the necessary electronics, plugged them in, and made a config file. The mechanical team also finished a drive module, so Andrew plugged it in to the REV Hub Interface on Coach Belbas' computer and test ran the module. It worked great. Ben programmed a basic teleop that allowed the gamepad to control the arm's bottom servo "Swivel". The teleop code allowed the arm to spin around in a 360 rotation, however we have to be careful due to the wire connections. If the arm spins around too many times the servo wires will be ripped out of the expansion hub, possibly breaking them.





Chassis Assembly (Mr. Bell, Mrs. Belbas, Zach, Mark): Today our goal was to complete the chassis assembly with wheel modules and the chassis baseplate. We needed to remove the supports from each wheel module side before assembling each module. Once we bolted each side together they would be ready to slide onto the chassis for us to get the exact measurement for each cross support. Sadly, we did not finish the whole assembly, but Mark will finish the assembly at home over the week. We were able to finish one module for the programmers to test. Zach drilled out the VEX mecanum wheels to enable the hex shaped metal reinforcements to slide into each wheel. These reinforcements will go through the onyx conversion pieces and fit tightly onto the axle to spin the wheel when the motor turns the axle.

Date: Sun, 10/24/18	***Location***: Tate House	***Purpose***: Programming Subgroup

Attendees: Ben

Agenda:
- Test the robot articulated arm by making all of the joints move & find the min and max servo positions

Reflections:

Mechanical/Programming (Ben): I tested the articulated arm with different weights on the end of the arm. I wanted to see if it could hold a weighted attachment for long periods of time.

------------------ Total

Weight: Observations about servo performance and items used in testing:

Test #10.0lbs I programmed the arm to move to each servos

Middle position. Then I set the arm to move based upon the gamepad values. The arm responded very quickly and smoothly.

Test #2.4lbs I added 1 wheel & 1 (25 tooth) gear that had a total weight of .4lbs. The arm moved to its starting position smoothly but didn't move as fast as the first test.

Test #3.8lbs Next I added another wheel and gear (weight =.4lbs both wheel and gear combined). The arm was very slow and was becoming unstable.

Test #41.2lbs I added 1 dc motor (weight =.8lbs) onto the end of the arm for this test. The arm was very unstable, when I tried to fully extend the arm it became unresponsive and froze in place a few inches from the table. I tried to lift the arm and the Shoulder servo died and dropped the arm onto the table.

This section is intentionally left blank.

Date: Thurs, 10/25/18	**Location**: ECAD, Inc.	**Purpose**: Autodesk Workshop

Attendees: Zach, Andrew, Ben, Coach Belbas

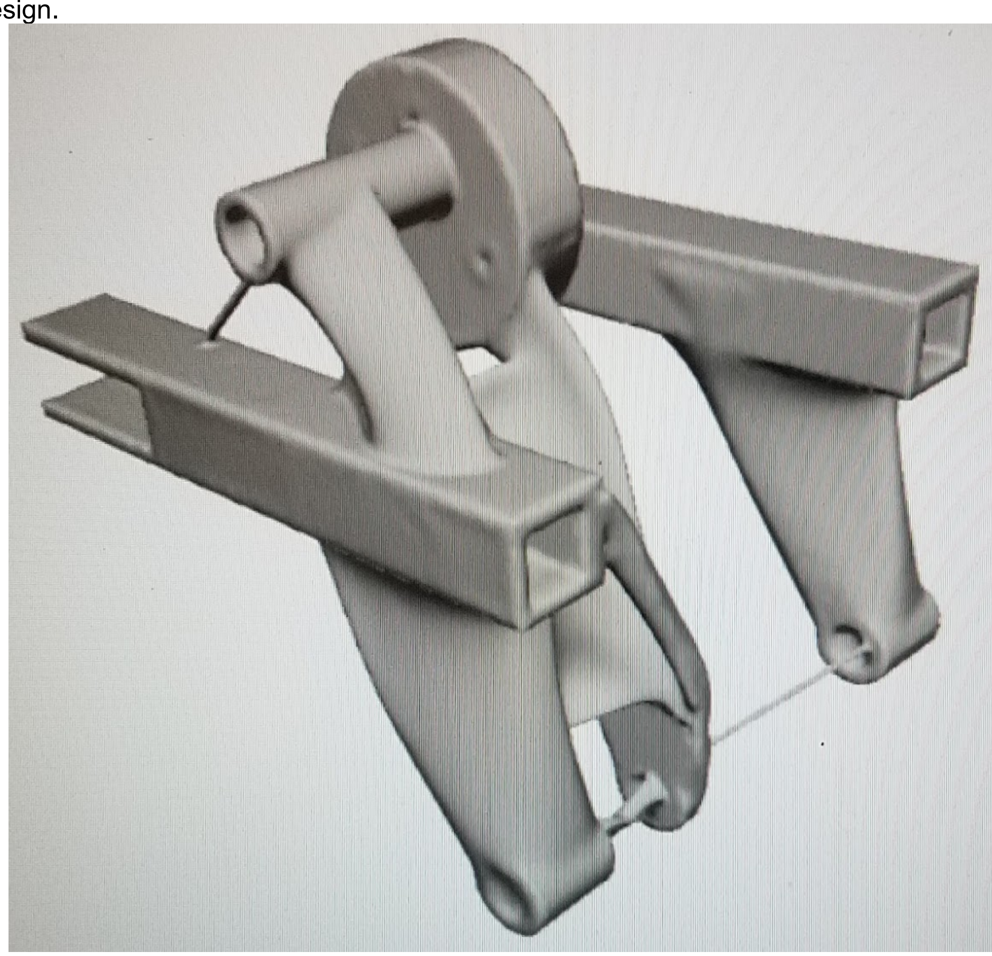
Agenda:
- Learn generative design from Autodesk professionals

Reflections:

Today we attended an Autodesk Workshop that was focused on generative design. It was taught by Mr. Ryan Abel and Mr. Allan Hsu, two Autodesk engineers. Generative Design was recently released in Fusion 360, the CAD program that our team uses. The advantage of using the generative design feature is that you can produce a part that has maximum rigidity with minimal mass. This is more cost effective, and it also streamlines the design process due to the built-in capabilities to program toolpaths directly from within Fusion 360 instead of having to export the files to a different software program.

Today we learned exactly how to create a part using generative design. First you define your design space. Then, you set the type of manufacturing you will use and specify your materials. Next, you define your loads and constraints. Finally, you let generative design solve for an optimal design using its pre-programmed algorithms based off of bone structure.

We also started using generative design on our wheel modules. We didn't finish, but we got valuable experience and instruction from the Autodesk professionals. We plan to use generative design on various modules once we get to later stages of their design.



Date: Fri, 10/26/18	***Location***: Belbas House	***Purpose***: Team Meeting #14

Attendees: Ben, Andrew, Luke, Joel, Mark, Zach, Coach Belbas

Agenda:
- Assemble ⅔ Field
- Work on Arm
- Fix Drive Modules

Reflections:
Andrew: Joel, Luke Ben and I put together a ⅔ field. After that I worked with Ben to bring over his arm teleop code into TeamCode and showed him more of how Source Tree works. I also removed the Nav-X gyroscope from Gar-E and put it in the new robot. Ben ran the Rev Hub Interface while Mark did sample tests of the prototype collector on the field.
Mark: Today, I worked on the drive modules. The main problem was too few of spacers in between the wheel and the inner side of the drive modules. Also, I started prototyping collecting mechanisms. First prototype was just a servo with zip ties on an axle and a salad box to act as a collection box.

Zach: Today, I cut each wheel cutout of the chassis one inch deeper. There had been a measuring mistake in CAD, and when we had put the wheel modules on, they exceeded the 18-inch limit. Then, we assembled and mounted the drive modules.

This section is intentionally left blank.

Date: Thurs, 10/25 - Sat, 10/27/18	**Location**: Lee House	**Purpose**: Programming Subgroup

Attendees: Andrew

Agenda:
- Finish Autonomous Architecture

Reflections:
- I added four child classes that inherit the main autonomous class. This lets us have four sets of unique movement and heading values, but still use the same state machine to step through the autonomous.
- Changed the drive method to be able to move the robot in any direction in autonomous:
  - Forward
  - Backward
  - Left
  - Right
  - 45 degrees in any direction
- I added a timeout feature to the drive method. This prevents the method from getting stuck and going forever, the method will time out and then the code will move on.
- I added a inches to encoder ticks converter to the drive method. This lets me type in how many inches I want to move and the method tells the motors to drive the equivalent in encoder ticks.
- I changed drive from a void return method to a boolean return method. Drive now returns a boolean on whether or not it is moving. If it is still moving, the method is not complete. If it is not moving then the method is complete.
- Added a telemetry object to the chassis class, allowing me to use telemetry calls in my autonomous methods.
- Tested and debugged the drive method. There were several logic error with drive where the code would seemingly skip the return statement and just go forever. I figured out that I was prematurely setting the method to return true before stopping the motors, meaning that the robot would drive forever.

This section is intentionally left blank.

Date: Mon, 10/29/18	**Location**: Belbas House	**Purpose**: Team Meeting #15

Attendees: Andrew, Ben, Mariah, Mark, Zach, Mr. Bell, Coach Belbas, Mr. Lee, Mrs. Tate

Agenda:

- Teach Mariah Wiring
- Work on Arm Code

Reflections:

(Ben): Programming work with Mentor Lee. Worked with the architecture and began creation of an custom Arm class for the Robotic arm. Arm class to be called ArmTeleop, created custom methods etc. for this class.

(Mariah): Today, Andrew taught me how to crimp wires and create a basic “data wire,” used to connect the sensors to the Rev hub.

(Andrew): I made a prototype phone holder with an acrylic cover to protect the screen. I will work on a protective REV case during the week. We need to make protective cases for the hub and phone so that the arm doesn’t bang up our electronics.

(Mark): Today, I continued to work on the collection box in CAD. We will 3D print this over the week and will assembly on Friday.

(Zach): Today I:

- Put our arm on the chassis to test for flipping. The chassis seems fairly balanced and did not flip.
- Switched out the aluminum channel with an aluminum tube to reduce the weight
- Weighed the minerals to see the weight we need to support on the end of our arm
- Switched out shoulder gears to get a 2:1 ratio. This gives us greater speed while reducing torque since we don’t need a lot of torque on the shoulder.

Date: 10/30/18	**Location**:Tate House	**Purpose**: Programming Subgroup

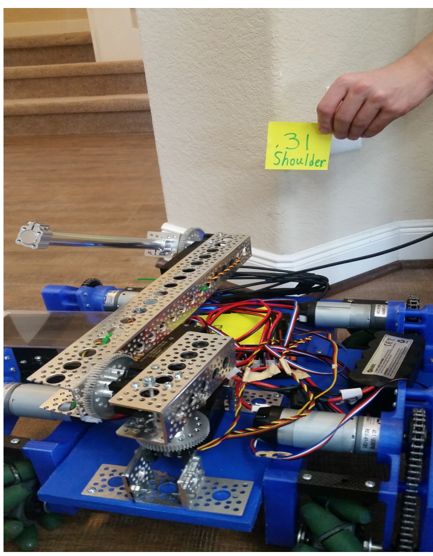
Attendees: Ben, Mrs. Tate

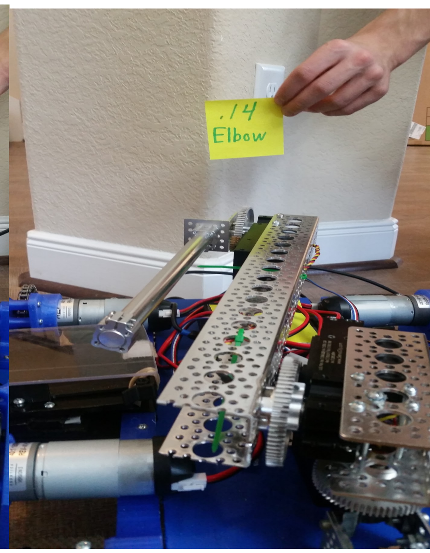
Agenda:
- Articulated Arm Programming

Reflections: (Ben)
Goal: Find positions for all 3 servos using Rev Hub Interface software on computer, connect gamepad, test

1) Found that the Team's expansion hub version is set to 1.7.2. The Secondary backup hub is set to 1.8.0. We might need to update Team Hub Firmware to the latest version so that we can use the same programs across expansion hubs.

2) Used Rev Hub Interface to locate ranged positions for all 3 servos. Documented the range positions of each servo.





3) Tested the Servos on the robot via gamepad and new ArmClass program. Noticed discrepancies with the setPositions. *researched the issue* -- After finding the values of the servo positions in the rev hub interface, I implemented them into the code. When we ran the program the servos went to a different position than they had with Rev Hub Interface.

4) I observed that a .5 value for the elbow servo will cause the servo to turn a full 360 degrees.

Date: Tue, 10/31/18	***Location***: Tate House	***Purpose***: Programming Subgroup

Attendees: Ben, Mrs.Tate

Agenda:

- Articulated Arm Programming

Reflections:

(Ben) Goals: Continue work for 3 servo positions; find max and min ranges. Setup a "Sleep" position for Robot where all servos are protected in a home position. Test on Robot.

**ISSUE from 10-30-18: Contacted Rev Robotics and asked about the servo setPositions being off from the RevHub Interface values. Rev Robotics said the issue was probably due to the particular servos our team is using and the the FTC class structure. They suggested there is something throwing the values off but he did not know what that was. This issue was discussed with team coach and she suggested we move forward and use the RevHub Interface to find starting values and then use the actual program and setPositions on the robot to find the real actual values the servos move to.

1)Tested elbow servo. Found "home" position issue. If rotating manually can reset the "home" at which the servo begins to rotate from. This will cause any setPosition to start from that new location.
Manual movement of the Elbow should be avoided.

2)Found "Sleep" positions for all 3 servos by trial and error on robot. Used prior day Rev Hub values as a starting point. Then added an offset. Photographed all 3 servo sleep positions for documentation.

3)Created a method armSleep that will send all 3 servos to their "sleep" positions. This will allow for safe driving and safe start location for all 3 servos during competition.

4)Tested sensitivity of the gamepad controls and how the arm reacts to driver inputs. Sensitivity at 1000 slowed everything too much, corrected to dividing by 100. This allows the setPosition to respond more naturally to driver commands.

5)During programming noticed a need for Telemetry Class functionality at the ArmClass level. This will allow for debugging the setPositions of the servos during testing phase. Currently there is no way to verify that the setPositions that are programmed are actually achieving the desired location.

6)Started creation of a buttonSleep method that will trigger the servo "sleep" positions by the driver. This would allow the drivers to send the servos to a resting position prior to the hard stop during competition. A hard stop cuts power and servos in a raised position will fall, possibly damaging parts. Could not complete this process due to missing functionality in the child class of ArmTeleop.

*ISSUE: When testing the articulated arm to get the proper set positions I ran into an issue when I manually rotated the arm towards the front of the robot,
the arm reset its starting position to its current position. The elbow servo should not be rotated manually to avoid a starting position reset.

Date: Thurs. 11/1/18	**Location**:Tate House	**Purpose**: Programming Subgroup

Attendees: Ben, Mrs.Tate

Agenda:

- Articulated Arm Programming

Reflections:

(Ben)

****THURSDAY- tried to understand limitations of servos. Lowest value was .005. When moving elbow found out that you can reset it's hardware origin. Found all forward values, had to find them all over again due to elbow position reset .Had to get new starting servo locations.

Looked at chassis and where the servos would hit, and got all 3 resting positions given the new hardware "origins". Had all 3 servos connected to gamepad worked on the sensitivity.

Realized 1000 was way too slow. Summary: gamepad responses not workable for accurate response times on servos.

PROBLEM: Arm would just drop when stopping program

SOLUTIONS: created Button press - coded for this. ARCHITECTURE ISSUE: Button functionality not present, had to bring down the references to the gamepad for this

ARCHITECTURE ISSUES: TELEMETRY STATEMENTS NOT AVAILABLE *** Huge programming testing issue, no way to send telemetry statements with current architecture.

SOLUTION: Raised issue with Coach and Team member responsible for architecture.

SUMMARY: A button added for end game protection of servo power cutoff - testing revealed gamepad inputs not usable with programming method algorithm -

standard debugging functionality and gamepad buttons missing in architecture - you can manually reset the original of the elbow servo causing all values to be reset.

This section is intentionally left blank.

Date: Fri, 11/2/18	**Location**: Belbas House	**Purpose**: Team Meeting #16

Attendees: Andrew, Ben, Mariah, Mark, Zach, Coach Belbas, Mr. Lee, Mrs. Tate, Luke, Joel

Agenda:
- Put all Electronics on the Robot
- Work on Arm
- Work on Collector
- Make a Battery Box for REV Batteries

Reflections:

Andrew: The highest priority task for today was to get all the electronics mounted on the robot. We need to put a REV Hub, a phone, a battery, a servo, and the team marker on the robot. I worked on modifying my prototype phone holder from the previous meeting to work. I ended up using a ninjaflex phone case from last year that we didn't use. I then velcroed a clear plastic cover onto the case to project the phone screen.

Joel: Luke and I worked on designing a new piece for the robot in CAD. This piece will both serve as a storage area for the battery and a flat surface above the battery on which to mount future attachments or the robot-controller phone. The photo displays the model upside down for clarity.

For the remaining meeting time, Mariah and I worked on mounting a simple servo to the back of the robot. This mechanism will serve as a way to deliver the team marker to the dropzone. Although rudimentary, this contraption will more than suffice for meet #1. This new attachment is named, "The Hitch."

Intake Module (Mark, Zach): Today, we first tested the collector with zip ties to see how they would collect the minerals. We discovered they were too slow and the zip tie "paddles" were too far apart, resulting in the minerals being collected very slowly.

We discussed replacing the zip ties with a brush, and then Mark had the idea of replacing them with one of the ninjaflex wheels that we custom-designed last season. We tested several different sizes:
- 2 ¾ inch diameter wheel
  - Balls: too tight
  - Cubes: too loose, barely sucked in
- 3 ¼ inch diameter with tape
  - Balls: too tight
  - Cubes: Sucked in with pressure and stayed in
- 3 ¼ inch diameter with tape
  - Balls: better, still too tight
  - Cubes: straightened out, then sucked in when pressure was applied
- 3 ¼ inch diameter with cuts in the wheel rim
  - Balls: better, still a little too tight
  - Cubes: didn't straighten out if crooked, sucked in with more pressure
- 3 ¼ inch diameter with a T-claw cut out in the wheel rim
  - Balls: a little tight but functional for meet 1
  - Cubes: sucked in with no pressure, straightened and sucked in with little pressure

Mariah: Today, I helped Joel make the mechanism that will drop the team marker, and learned how to organize the electronics on the robot. I organized all of the wires for easy access, and got the electronics onto the chassis.

We also discussed adding a spring-loaded plate to the bottom of the collector box, which would function to push the minerals back to the ninjaflex wheel to be ejected from the intake module into the cargo hold, similar to how a grocery store shelf pushes items forward on a shelf with a spring-loaded plate.