Design Project 1 - Autonomous Sumo Battle Robot

Design Team 5
Cornerstone of Engineering I : GE 1501
Benjamin Adler, Megan Anderson, Dominick Doyle, and Anjali George
December 7, 2023

B. Adler, M. Anderson, D. Doyle, A. George
Design Team 5
Northeastern University

Professor R. Whalen
Northeastern University
Boston, MA 02115
December 7, 2023

Dear Professor Whalen,

For this design project we were expected to utilize the engineering design process in order to construct an autonomous robot to compete in a Sumo style competition. During the competition, the robot would be placed within a Sumo ring directly across from an opposing robot. The goal was to create a robot with designed components, both hardware and software, that was capable of keeping itself within the ring whilst also attempting to force the opponent out. Another objective that we identified was to take into account possible modifications that opposing teams may make to their robots when designing our prototype in order to improve our likelihood of succeeding in the competition.

After hours of brainstorming, we came up with components that we wanted to prioritize incorporating into our robot, as well as some initial sketches of possible designs. When considering what we wanted our robot's main attack mechanism to be, we debated ideas such as a wedge, a claw, a hammer, a rotating notched wheel, and even a forklift style weapon. Our decision to incorporate a wedge style ramp into our final design was a direct result of its historical use and effectiveness in Sumo robot battles, as well as its simplicity. When looking into how we were going to address the possibility that opposing robots were also going to have their own weapons, we wanted to incorporate a design that protects our hardware from damage. To accomplish this we came up with ideas such as selective armor, skirts, a shell, and even just a simple cover for our exposed top wires. In the end we decided that a full protective shell would be the best option as it keeps everything protected and successfully combats virtually all of the opponents possible attack mechanisms.

For our prototype, Yoshi, we incorporated a snow-plow style wedge attached to the front of the metal chassis. On top of this, we settled on a full body cardboard shell instead of our original 3D printed design after motor issues arose late in the process as a result of the heavy weight. We 3D printed a battery case which was then attached to the bottom of the chassis in order to lower the robot's center of mass. We used two ultrasonic distance sensors, embedded within the front ramp to improve the tracking capabilities of our robot. With these sensors the robot gained the ability to locate robots, lose sight of them, and recalibrate itself in order to locate the robot again and resume attacking. The two light sensors on both front corners of the chassis worked to sense changes in light and were the sole reason our robot was able to remain in the circle autonomously. During design we faced numerous challenges that forced us to quickly adapt in order to move forward. From these challenges we learned that the engineering process truly is a circle and believe we all left as better engineers with an increased understanding of design, as well as new skills to add to our engineering toolbox.

Sincerely,
Design Team 5

Table of Contents:

List of Figures:

List of Tables:

Definitions:

KTPA: *Kepner-Tregoe Problem Analysis; a matrix that assists in revealing the root cause of an engineering problem*

Duncker Diagram**:** *a creative tool that can help refine an engineering problem*

FYELIC: *First Year Engineering Learning & Innovation Center*

PLA: *a vegetable-based plastic material used in 3D printing*

Abstract:

Autonomous sumo robot competitions, while particularly popular in Japan, have been taking the United States by storm in recent years. These competitions consist of two robots battling head to head on the chosen sumo ring, with the robot that remains within the ring for the longest being declared the winner. The robots use a wide variety of sensors and motors in order to function autonomously. Sumobots require extensive programming of this hardware, as well as recurring application of the engineering design process, in order to function at a high-level. For this study, the purpose is to design our own sumo robot in four-person design teams to compete in an in-class competition. Our robot will look to address errors in previous designs, enhance autonomy through complex code, and display a high-level performance during the competition using a wide variety of hardware and software. Current sumo robots utilize “weapons” and “armor” to assist in pushing the opposing robot outside of the circle while simultaneously being protected. While there are many weapon options, we found that the wedge technique is one of the most common, and appears to be most effective at hindering the opponent's ability to remain inside the circle. Our robot utilizes a front ramp, a variation of the common wedge, as our main weapon. This design also incorporates a full protective shell acting as armor to keep all the hardware safe despite the impact faced during battle. In order for the robot to function, a variety of light and ultrasonic sensors are used to communicate and direct the two 360 degree servo motors on the back of the robot. The Sumo 2 competition granted virtually full design freedom, with the only constraints being the size (16in x 12in) and weight (3.5lbs) of the robot. Our robot was successful, it was able to remain within the sumo ring and attack opposing robots completely autonomously. Our placement in the competition was lower than we would have liked (6th place); however, we have linked this to minor hardware malfunctions, which we would look to address in future research. This research is extremely relevant for those participating in future sumo robot competitions, as it contains valuable information, tips, and strategies for the design process of an autonomous sumo battle robot.

Problem Definition:

Going into the Sumo 2 phase, the first step was creating a refined and logical problem statement that encompassed each of our newfound goals and constraints of the competition. We achieved this by utilizing problem definition methods including a Duncker diagram (Fig. 1 below) and a KTPA matrix (Table 1 below).

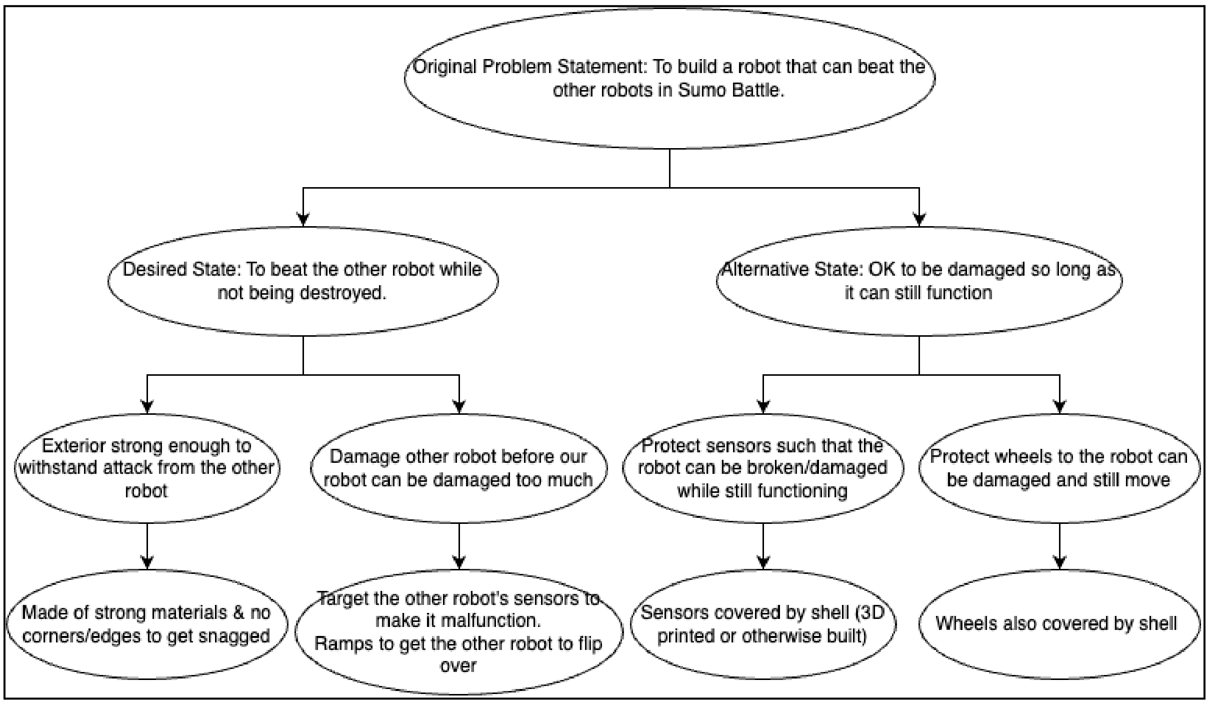


Figure 1, Duncker diagram problem definition; The diagram outlines various pathways and creates a variety of solutions, even ones that do not meet desired state. This Duncker diagram helped identify areas of priority for our robot moving into the design stages (Team 5, 2023)

	Is	Is Not	Distinction	Possible Cause
What	Robot to attack opponent robots	Robot that only defends itself	Robot should utilize offensive AND defensive strategies	Offensive v. defensive is determined by programing strategies
Where	Within the circle	Outside the circle	Robot must stay inside the circle and avoid being pushed out	Being pushed out of the circle by opponent, sensor malfunctions, not sensing edge

When	While being attacked by opponent or while opponent is sensed	While being attacked by opponent	Robot should have offensive strategies and “hunt” other robot	Offensive v. defensive is determined by programing strategies
Exent	Withstand opponent (either until it leaves the circle or stops functioning)	Withstand opponent	Only defensive robot relies on the other robot malfunctioning by itself	Offensive v. defensive is determined by programing strategies

Table 1, KT problem analysis matrix; This matrix was formulated to help identify possible causes of problems faced during Sumo 1 (Team 5, 2023)

By using these problem definition methods, we were able to establish that the key goal we needed to address was to design and build a robot that is durable enough to withstand any attacking force from the opponent robot and uses offensive strategies to scan and attack its opponent. The primary objective of the robot is to push the opponent out of the ring. This is achieved through the functions of the robot which include remaining inside the circle, evading damage from the opponent robot, and scanning using the distance sensors to seek out and attack the opponent. The constraints of the robot include size (12 x 16 in) and weight (under 3.5 lbs) requirements. From this definition stage, we were able to create a refined problem statement as follows:

Problem Statement: Design and create an autonomous robot that stays within the designated white ring on the sumo mat while also seeking out opponent robots to push out of the circle while remaining in bounds. Account for protection of internal wires and sensors, and increase speed in comparison to Sumo 1 robot in Sumo 2 robot design while also meeting weight constraint of 3.5 pounds and size constraints of 12 inches by 16 inches.

Figure 2, Problem statement; Refined problem statement for the sumo 2 competition (Team 5, 2023)

Background Research:

Prior to the ideation and generation stage of Sumo 2, we did a significant amount of research into past Sumo robot competitions, tips and tricks from various robotics sources, as well as looking into the specifications of the parts we intended to use. In our initial research, we looked into videos of other Sumo competitions to gather intel on what works well in competition and what does not. From this research, we found that robots with some sort of wedge or ramp tend to perform better in competition as they are better able to flip or disorient their opponent in some way, shape, or form. Additionally, we noticed that robots that utilized a circular scanning strategy were able to seek out and attack their opponent more efficiently than those that used a linear default movement.

In addition to videos, we utilized various articles that discussed Sumo robot competitions and the design specs of various parts in order to generate ideas for the best strategies. First, we found that maximizing the torque of the motors is ideal in order to be able to handle the weight of the opponent if pushing or being pushed. This was important in our considerations regarding motor choice for our wheels. According to the user guide and patent of the Parallax servo motors, when fully powered the motors provide 38 oz-in of torque. However, the Sparkfun DC motors only provide 800 gf-cm or roughly 11 oz-in of torque. This comparison aided our decision to stick with the servo motors for the Sumo 2 competition.

Our group also did significant research into the importance of friction, momentum, and the location of the center of mass of the robot in competition success. We found that a higher wheel friction reduces the opponent's ability to push the robot. Friction is associated with the weight of the robot and the traction of the wheels, both of which should be maximized in order to remain as stationary as possible when pushed. Additionally, an increased momentum increases the robot's ability to move the opponent and decreases the opponent's ability to move the robot. Momentum is associated with the mass and velocity of the robot. Therefore, maximizing weight and speed are ideal. Lastly, a lower center of mass also decreases the opponent's ability to push our robot. These factors were all vital in our idea generation and decision making processes.

The last significant form of research we did was to address the speed of our robot. However this proved slightly more difficult than expected as we aimed to keep the servo motors due to its optimal torque as previously discussed. The servo motors are programmed to go one speed, therefore we sought to increase the radius of the wheels to increase the linear speed of the robot with the constant rotational speed. In order to find the most optimal radius we again consulted the Parallax motor design specifications to find the rotational speed of the servo motors (50 rpm).

Design Generation:

In order to generate solutions to our identified problems, we utilized a few key generation methods. First, we used brainstorming and inversion techniques to come up with a variety of initial ideas. We wanted to gather a large range of ideas and ways to creatively address the problems, so we spent multiple team meetings sharing and expanding on different ideas. We also used the inversion technique to address the flaws and areas of improvement of Sumo 1, in order to determine solutions that would prevent those flaws. For example, we noticed that edges and corners of the robot tended to get caught on the opponent and ultimately were counterproductive in achieving the objectives of competition. Therefore, we determined early on that some type of shell or protection with minimal areas for an opponent to get caught would be ideal in our design.

After coming up with a range of ideas, we created sketches (Fig 3. below) that each implemented some subset of these ideas. These sketches included different variations of protection for the internal wiring of the robot as well as different offensive strategies including a curved outer edge to latch on to opponents, a light that would trigger opponents' sensors keeping them away from the body of the robot, and multiple ramp variations.

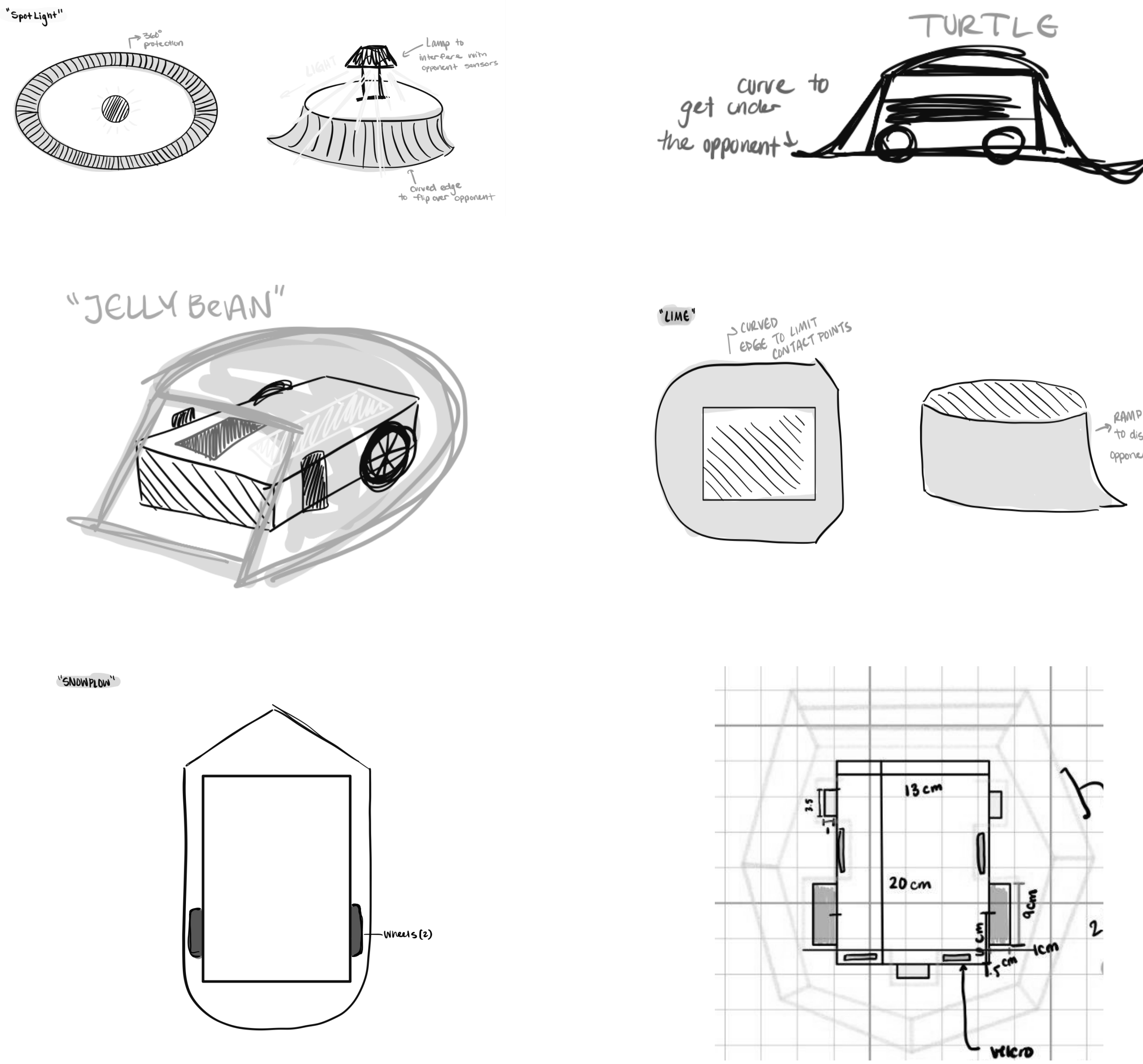


Figure 3, Brainstorming sketches; Initial drawings formulated in the idea generation stage of Sumo 2 (Team 5, 2023)

These drawings were worked on collaboratively so each team member was able to add components that they believed were important in facilitating our robot's success. Each individual sketch contained unique elements that collectively contributed to our initial design. Through the brainstorming process, we were able to successfully develop potential solutions that later aided in constructing our final design. The overlap between each sketch highlighted the priority of a full protective shell and a wedge/ramp as the main attacking mechanism. The team agreed that these elements, if implemented properly, would give our robot the best chance to succeed in the competition.

Evaluating Design Alternatives:

After coming up with a variety of ideas, we needed to narrow down and decide which design would best suit our problem statement and ultimately achieve the best results in competition. In order to accomplish this, we utilized a rank order comparison table (Table 2 below) to determine the importance of each function and a decision matrix (Table 3 below) to determine which design best achieved the ultimate objectives.

Objectives	Shell/ Protection	Wheel Traction	Speed	Power	Weight	Ramp	Code/ Strategy
Shell/ Protection		.5	1	1	0	1	1
Wheel Traction	.5		.5	.5	.5	0	0
Speed	0	.5		1	1	1	0
Power	0	.5	0		.5	0	0
Weight	1	.5	0	.5		1	.5
Ramp	0	1	0	1	0		0
Code/ Strategy	0	1	1	1	.5	1	
Totals:	**1.5**	**4**	**2.5**	**5**	**2.5**	**4**	**1.5**

Table 2, Rank order comparison; Created to help identify the most important objectives to focus on in design and development (Team 5, 2023)

Through the use of rank order comparison, we were able to identify which objectives were most important. We determined that the ramp, power, and traction were the most crucial aspects, so we centered our designs around these objectives. Our main focus was the shell, as it was multifunctional, encompassing multiple design objectives and functions, specifically the protection and weight categories..

Following this, we wanted to highlight our robot's power and the ramp, as these two objectives combined would make significant advances in our robot's offensive capabilities. Ranking these objectives made it significantly easier to prioritize the robot's features as we moved into the decision phase.

Objective	Shell/ Protection	Wheel Traction	Speed	Power	Weight	Ramp	Code/ Strategy	Totals
Weights	1.5	4	2.5	5	2.5	4	1.5	
Design 1: JellyBean	8/12	5/20	5/12.5	7/35	9/22.5	7/28	X	**130**
Design 2: Turtle	8/12	5/20	5/12.5	7/35	9/22.5	4/16	X	**118**
Design 3: Snowplow	9/13.5	5/20	5/12.5	7/35	9/22.5	3/12	X	**115.5**
Design 4: Spotlight	7/10.5	5/20	5/12.5	7/35	9/22.5	5/20	X	**120.5**
Design 5: Lime	7/10.5	5/20	5/12.5	7/35	9/22.5	6/24	X	**106.5**
Design 6: Yoshi	9/13.5	8/32	9/22.5	7/35	9/22.5	9/36	X	**148**

Table 3, Decision matrix; Made to help determine the best course of action in the Sumo 2 build (Team 5, 2023)

During the generation phase, we formulated various design ideas and ranked them based on how each design satisfied our outlined objectives using a decision matrix (Table 3 above). Since the code we developed for Sumo 2 was independent of the shell design, we did not include it in the decision matrix totals. Additionally, since the power, speed, and traction of the robot were based on factors independent of shell design (motor choice and wheel design), each design was given the same rating for those factors. Lastly, each shell design would give the robot approximately the same amount of additional weight. Therefore, each shell design was also given the same rating for that factor. Based on this matrix and the feasibility of creating each design, we identified that the Yoshi robot design was the best suited design to meet all of our design objectives.

Based on the result of our decision matrix, we began the transition from generation to implementation. This included the creation of to-scale AutoCAD drawings (Fig. 4 below) and final sketches of the shell design (Fig. 5 below).

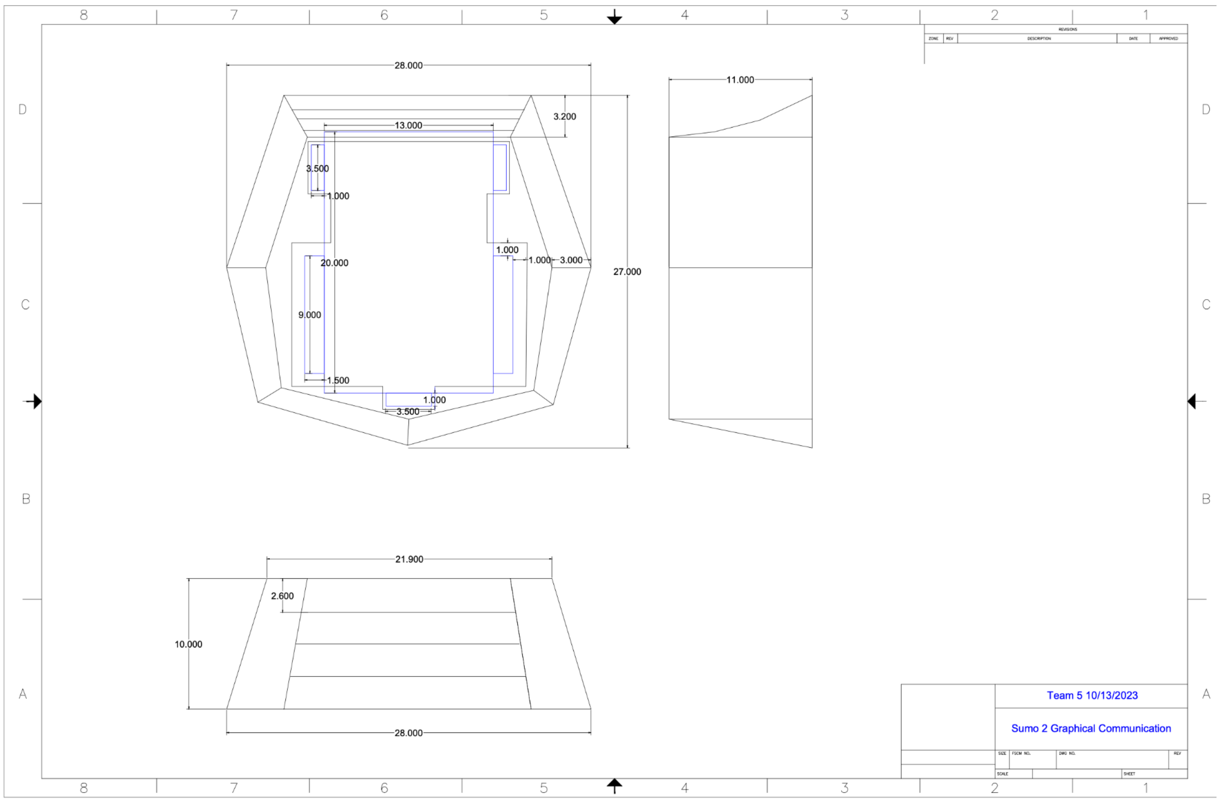


Figure 4, Shell AutoCAD drawings; Made to help translate rough sketches into a to-scale drawing that could be implemented and 3D printed (Team 5, 2023)

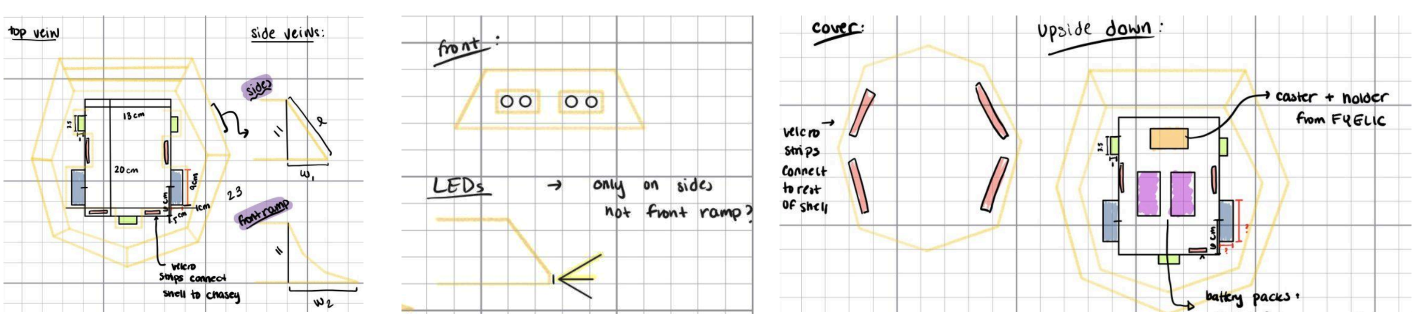
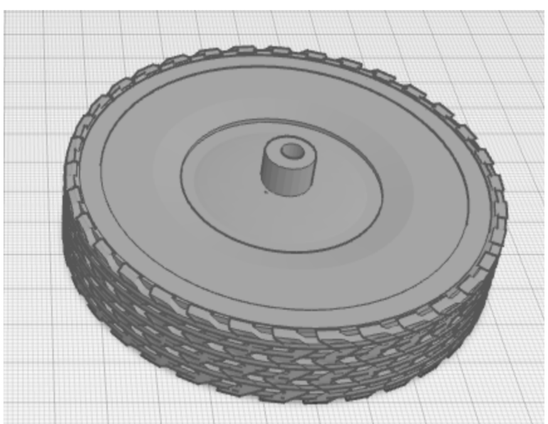


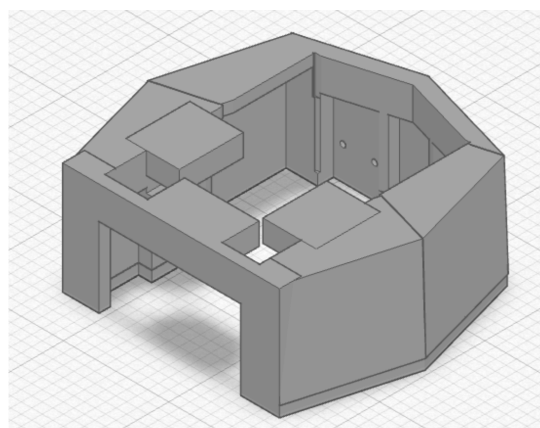
Figure 5, Final design sketches; Made to help translate rough sketches into to-scale drawings that could be implemented and 3D printed (Team 5, 2023)

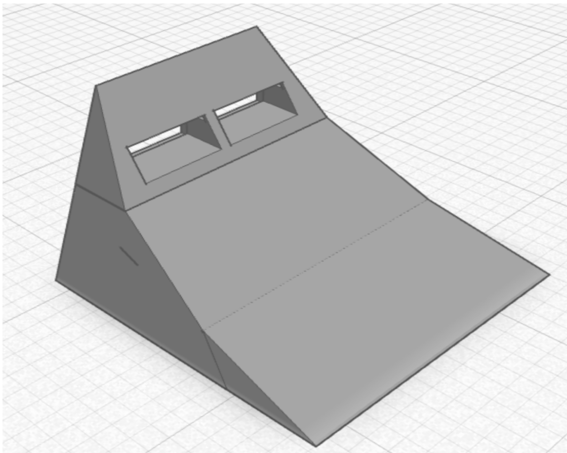
These final design steps were vital in ensuring that our initial ideas would fit competition requirements and that each piece would fit accurately with the others. The team agreed on these final sketches and dimensioned drawings, and we began to move into the implementation phase of the design process by creating our initial prototypes.

Implementation & Iteration:

In implementing our design ideas, while many of the created prototypes were successful, we came across a few key problems that affected our final product. We initially moved forward with our 3D printed designs which included the shell, ramp, wheels, and battery case. Each of these prototypes were originally modeled using TinkerCAD (Fig. 6 below) and created using predetermined measurements that would both fit the body of our robot as well as the size constraints of the competition.







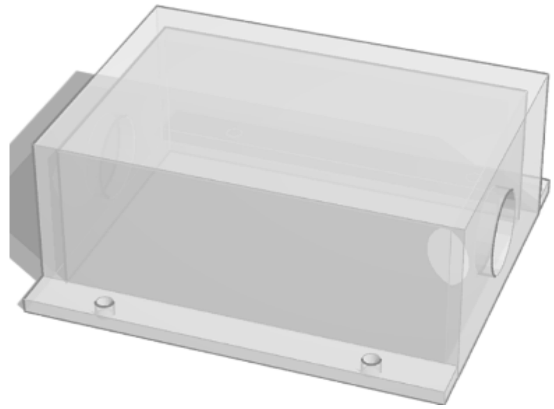


Figure 6, 3D models; 3D modeling of robot wheels, shell, ramp, and battery case to be printed. Models were designed using predetermined measurements to fit the chassis in the software TinkerCAD (Team 5, 2023)

In this initial implementation stage, each aspect of the design addressed the objectives we previously decided on in the generation phase of the design process. First, our ramp modeled multiple objectives including weight, front protection, disorienting the opponent, and interfering with opponent sensors. The ramp added to the total weight of the robot which would increase its ability to withstand pushing from other robots. Additionally, it provided front protection specifically for our distance sensors and would hinder other robots if they drove up the slope. Lastly, we intended to print the ramp with white PLA which would trigger the opponent's sensors and cause them to back away from our robot. After designing our first prototype using the 3D modeling software, we 3D printed the shell at FYELIC and went through multiple rounds of testing to ensure its sturdiness, dimensions, and clearance to the floor. After this testing, we determined that this ramp fulfilled the intended objectives and it ended up being used in our final robot for competition.

Similarly, the new wheels we designed also addressed multiple intended concepts including speed and traction. With the increased radius compared to that of the original wheels, these prototypes would allow our robot to move faster than it did during Sumo 1 (Appendix 2). After 3D modeling, we 3D printed the wheels using resources at EXP. Since we used resources outside of FYELIC to 3D print, this was one of the design aspects that required purchase (Appendix 1). In addition to paying for printing, we also purchased rubber cement to coat the edges of the wheels (Appendix 1). This addressed the traction that was lost in the transition from the original Sumo 1 wheels to the 3D printed plastic. After printing and coating the wheels, we completed multiple rounds of testing that ensured the traction, fit, and durability of the wheels, as well as ensuring that the rubber cement did not break any Sumo rules (ie. being able to pick up and hold an index card for over two seconds). After this testing, we determined that the wheels fulfilled the intended objectives and they were used in our final prototype.

The battery case we designed mainly addressed the idea of creating a lower center of gravity for the robot since it was intended to hold the batteries underneath the chassis rather than on top. After the design of the 3D prototype, we printed the battery case at FYELIC and ensured its dimensions fit the battery packs as well as the grills in the chassis. After this testing, we determined that the case fulfilled the intended objective and was used in our final prototype.

The most time and labor intensive aspect of our initial design was the shell. The shell was designed in order to address the goals of weight, protection for internal hardware, and decrease the opponent's ability to latch onto our robot. While the original design met all of these objectives, after 3D printing the prototype and attaching it to the body of the robot, we encountered issues with the weight objective. The shell was able to bring the total weight of the robot to the competition constraint of 3.5 pounds, however this prototype disproved our original idea to build to the maximum, as the motors were unable to handle that much weight. During testing of the shell, we noticed there was a significant decrease in speed of the robot as well as drag caused by the shell which decreased the robot's overall mobility.

These observations forced us to revisit previous steps of the design process in order to come up with a solution that provided protection for the robot, without sacrificing speed. We went back to the ideation stage in order to come up with a new, lightweight material to fabricate the shell out of. After considering the options available to us in the short time left before the competition, we decided to create the new shell out of cardboard. This material provided a relatively sturdy solution that was able to effectively protect the robot's internal hardware.

The final aspect of implementation of our designs was creating a code that would be able to utilize our added features. The robot's coding strategy took an offensive approach and prioritized attacking opposing robots first. This version of the code (Appendix 4) truly highlighted the autonomous capabilities of the robot. Using a series of "if" tests centered around the states of the front ultrasonic distance sensors and the side/back light sensors, the robot was able to accomplish our competition goals. This code accounts for four of the robots primary states during battle.

The first of these states, the scan, would occur when no opponent was sensed within 20 inches of either front distance sensor. In this state, the code would direct the servos to spin in opposite directions, rotating the robot accordingly until a robot was picked up by either of the front sensors. Once the enemy was sensed, our robot would enter the second possible state, recalibration mode. In this mode, our robot would automatically spin in the direction of the opponent based on what front sensor was hitting. The robot would then continue to spin, centering itself with the opponent, until both sensors were reading the parametrized distance. This would lead the robot to enter attack mode, the third and final offensive state. Under these conditions, the servos would both spin forward, propelling the robot toward the opponent, attempting to force it out of the ring.

The last state that this code encompassed was perhaps the most important state as far as facilitating our robots' success. This section of the code utilized the side and back light sensors, as well as their states, to keep our robot within the ring. Under this section, our robot would spin the servos according to what sensor was hitting the outer white edge of the ring, in order to prevent the robot from leaving the circle, resulting in a competition loss. These components combined all enabled our robot to successfully pursue opponents and remain within the circle autonomously.

Evaluation:

Our main design objectives during Sumo 2 were speed, power, protection, and traction. Yoshi accomplished the speed design goal through our improved 3D printed wheels which enabled the robot to cover a greater distance and move at a higher speed due to the increased wheel radius. Our design successfully addressed our power objective by using servo motors rather than the Sparkfun DC motors, as the servos have more torque. The final cardboard shell addressed the protection design objective by shielding Yoshi from all angles, covering the side sensors, internal wiring, and wheels from potential interference from the opponent. The 3D printed ramp we created also addressed our protection objective by redirecting the opponent from hitting our robot head on, and served as an offensive tool. In order to meet our traction design objective, we used rubber cement to coat the 3D printed wheels. This created a tackier surface on the wheels and greatly increased the traction to help prevent Yoshi from being easily pushed or sliding out of the circle.

The design of our robot on competition day and the original prototype proved to be drastically different. Our original sketches of the design that we planned to implement incorporated the 3D printed full body protective shell. However, as we approached competition day we discovered that our original 3D printed shell would hinder our robots ability to move effectively due to the motors being unable to carry the total combined weight of the shell, body, battery pack, and ramp. This setback demanded that we adapt quickly, and move back to the generation phase to develop a new solution that addressed our protection objective. We designed an entirely new shell, constructed out of cardboard, that was durable enough to protect Yoshi while enabling the robot to move at optimal speed due to its lightweight nature. Originally, we proposed the idea of using light to interfere with opponents' sensors, however this idea was scrapped after our team realized the complexity of powering the LED’s within our time constraints was not feasible. The rest of the robot remained consistent with our original sketches, as it implemented the two-sensor plow and multi-sensor application.

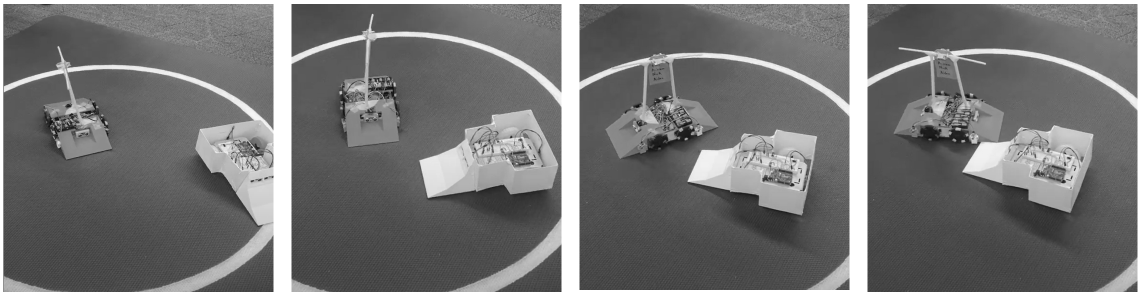


Figure 7, Yoshi in Action; Taken during competition to display the code strategy of the robot. (Team 5, 2023)

Our final prototype, Yoshi, was able to accomplish all of our intended design goals. The robot successfully resisted force from the opponent robot, was not pushed out of the ring, and did not take any damage to the sensors, wires, or wheels. However, during the competition we discovered that the motors were not functioning at full capacity, as they would intermittently lose power due to wires connecting the servo motors to the breadboard being slightly pinched by the side of the cardboard shell. Although our competition placement was not ideal, it did not reflect the full capabilities of our robot as all other aspects of the design were fully functional.

Reflection:

Upon reflection, this project has allowed us to gain first-hand experience in the circular nature of the design process and the multitude of trials, tests, and revisions often required before reaching a final solution. In order to create the most optimal designs in the given time constraints, we prioritized the idea generation and prototyping phases, and attempted to consider all obstacles and issues that might occur. We designed multiple models out of cardboard that demonstrated our design ideas for the wheels and shell. Additionally, we considered the necessary dimension and weight constraints in our finalized design. Throughout this process we were able to solidify our design choices and run tests to ensure that our design was functional and met our objectives. However, despite these precautions, we still faced obstacles in the final steps of implementation that forced us to revisit previous steps in the design process.

While initially frustrating, this experience allowed us to gain first-hand experience in the non-linear paths that engineers must take in order to generate a final solution.

This project also taught us the value of failure and the opportunity it provides to grow, work towards a better solution, and strengthen our knowledge to ensure we don't make the same mistake twice. Through the process of trial and error, we were able to gain a better understanding of how to code with Arduino IDE, work with Sparkfun, develop designs on TinkerCAD, 3D printing, and more.

If given more time and resources, we would find an alternative material to construct the outer protective shell, such as wood or laser-cut acrylic, which would provide a more durable but lightweight alternative to our 3D printed shell. Additionally, we would add more clearance to the sides of the shell to avoid any wires from being pinched and allow the motors to function at full capacity.

In the early stages of this design project, we played with the idea of a robot that produced mutually assured destruction. This included creating obstacles and offensive strategies that would ensure the damage of the other robot's sensors and functionality while risking the functionality of our robot. While this idea was intended to limit destruction and damage to the robots themselves, these ideas do raise ethical concerns when viewed from the perspective of larger applications. As seen throughout history and in the development of many dangerous technologies, if extended beyond a competition with stakes as low as that of the Sumo competition, these ideas would be ethically ambiguous at best.

Despite these initial concepts, our final robot did not follow the ideals of mutually assured destruction, therefore not causing any major ethical concerns to humanity. The robot was incapable of reaching speeds dangerous to humans. Additionally, it was not weaponized in any way that would cause bodily harm. Overall, we did not violate any major engineering codes during the design process of this project.

While this project addresses society on a small scale, our sumo bot provided a service to humanity by creating a learning experience for all of our group members. Through creating Yoshi, we were able to learn more about how to utilize Arduino IDE, SparkFun, and more. The experience we have gained will allow us to become more skillful engineers to potentially benefit humanity in the future.

Bibliography

[1] Tycho, “Sumobot Tips and Tricks – Iron Reign Robotics,” *ironreignrobotics.org*, Apr. 12, 2016. https://ironreignrobotics.org/2016-04-12-sumo-tips/

[2] Hino, “Mr. Hino’s 3 Strategies To Win A SUMO BOT Competition!,” *www.youtube.com*, 2019. https://www.youtube.com/watch?v=tw7HErc4z8k (accessed Dec. 06, 2023).

[3] J. Slifka, “Lego EV3 Robot Sumo Wrestling BattleBots Challenge,” *www.youtube.com*, 2015. https://www.youtube.com/watch?v=suyPkO0apak (accessed Oct. 15, 2021).

[4] IctProfi, “10 Perfect Sumo Fights(INFOMATRIX-Romania) - Lego EV3 Robot Sumo Wrestling BattleBot Challenge,” *YouTube*. Jun. 29, 2017. [YouTube Video]. Available: https://www.youtube.com/watch?v=JNbLaEFPe4k

[5] K. Al-Olimat, B. Wilson, and T. Germann, “Sumo Robot Competition,” *Sumo Robot Competition*, 2016.https://asee-ncs.org/wp-content/uploads/2021/12/proceedings/2016/student_regular_papers/2016_ASEE_NCS_paper_58.pdf

[6] “Parallax Continuous Rotation Servo - Downloads - Parallax,” *www.parallax.com*, 2023. https://www.parallax.com/package/parallax-continuous-rotation-servo-downloads/

[7] “Hobby Gearmotor - 140 RPM (Pair) - ROB-13302 - SparkFun Electronics,” *www.sparkfun.com*, 2023. https://www.sparkfun.com/products/13302

Appendices:

Appendix 1 - Bill of Materials

Date (m/yr)	**Material**	**Cost**
10/26	Plywood (12in x 12in x 1/8in)	$5.29
11/13	Wheels 3D Print	$3.93
11/14	Rubber Cement	$2.50
11/27	White Spray Paint	$10.61
		Total Cost : $22.33

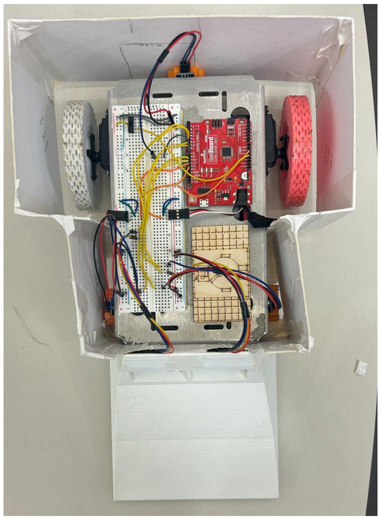
Appendix 2 - Speed Calculations

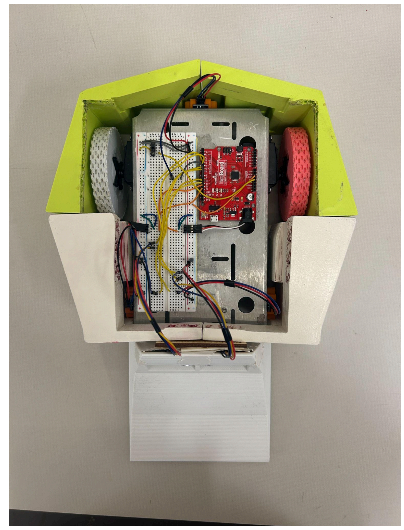
$$v_t = \omega r$$

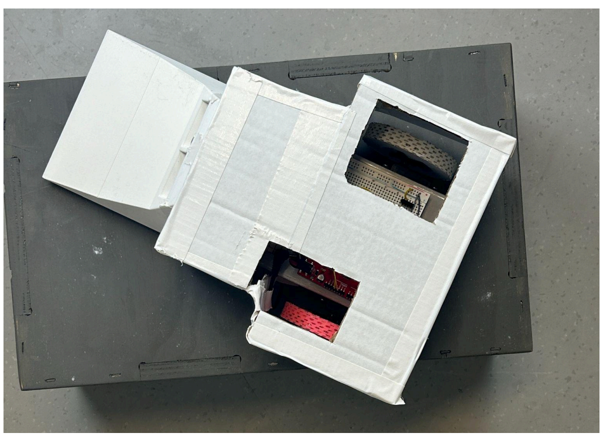
Rotational Speed: $\omega = 4.75\ rad/s$

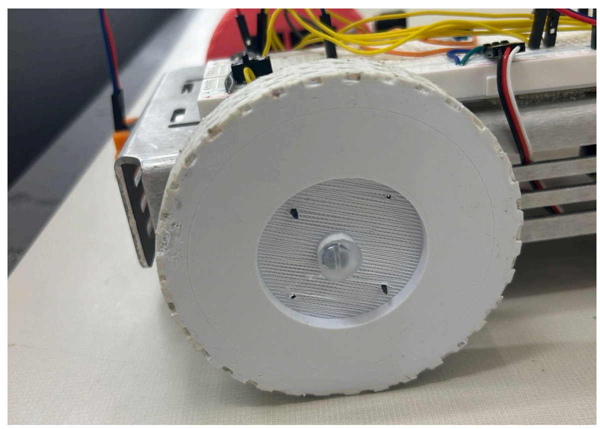
RADIUS (cm)	SPEED (cm/s)
3.5	16.7
4.0	19.0
4.5	21.375
5.0	23.8
5.5	26.125
6.0	28.5

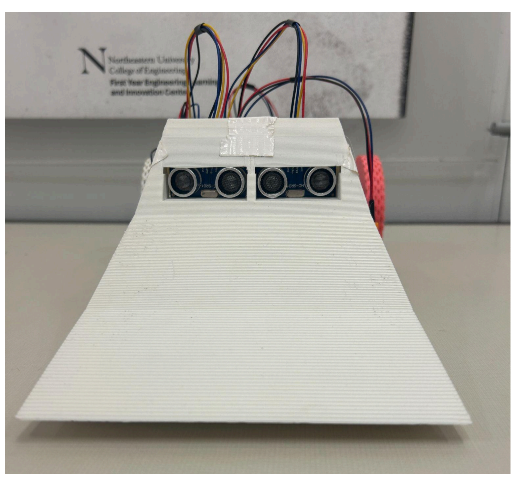
Appendix 3 - Additional Photos











Appendix 4 - Code

/* Written by: D.Doyle, M.Anderson, A.George, and O.Adler 11/18/2023
doyle.do@northeastern.edu

```
Purpose: Autonomous sumo robot control with Sparkfun redboard for final sumo project check
 This program functions as the code for our autonomous sumo robot for the second battle
 This program utilizes two side sensors and their states as well as a back sensor to keep the
robot inside of the circle
 This program utilizes two front USD sensors in order to be able to compleltely sense and
adjust to where the opposing robot is
 When a robot is not in range the servo motors will spin in opposite directions, dependent on
the phase, spinning/scanning the robot
 When a robot is in range, dependent on distance, the servos will spin in a certain direction
putting the robot into the ideal attack mode

*********************************************************************************************/

#include <Servo.h> //servo library

Servo servo_R; //right servo
Servo servo_L; //left servo

const int leftSensor = 11 ;   //assign sensor pins for three down light sensors
const int rightSensor = 10 ;
const int backSensor = 9 ;

const int trigPinright = 7 ; //assign pins for trig and echo pins for first usds
const int echoPinright = 6 ;

const int trigPinleft = 5 ; //assign pins for trig and echo pins for second usds
const int echoPinleft = 4 ;

int phase = 0; //initialize boolean varibale
int lsensorState = 0; //initialize sensor variable left
int rsensorState = 0; //initialize sensor varable right
int bsensorState = 0; //initialize sensor variable back

float distanceright = 0; //stores distance measured by usds for right
float distanceleft = 0; //stores distance measured by usds for right

int flag = 1;

void setup() { //do setup once
//---------------------------------
  pinMode(leftSensor, INPUT); //declare assigned sensor pins as input
  pinMode(rightSensor, INPUT);
  pinMode(backSensor, INPUT);

  servo_R.attach(12); //servo attachment pins (...) for numbers of pints to be input)
  servo_L.attach(13);

  pinMode(trigPinright, OUTPUT);  //USDS pins trig outputs, echo recieves RIGHT SENSOR
  pinMode(echoPinright, INPUT);

  pinMode(trigPinleft, OUTPUT); //USDS pins trig outputs, echo recieves LEFT SENSOR
  pinMode(echoPinleft, INPUT);
}

void loop() {
//---------------------------------------------------------------------------------
  rsensorState = digitalRead(rightSensor); //right sensor read
  lsensorState = digitalRead(leftSensor); //left sensor read
  bsensorState = digitalRead(backSensor); //back sensor read
```

```
  distanceright = getDistanceright(trigPinright, echoPinright) * 1.0051; //get distance from
right sensor and assign to right distance variable
  distanceleft = getDistanceleft(trigPinleft, echoPinleft) * 1.0051; //get distance from left
sensor and assign to left distance variable

  Serial.print(distanceright); //print distance from right sensor
  Serial.println("in"); //print units (in)

  Serial.print(distanceleft); //print distance from left sensor
  Serial.println("in"); //print units (in)

//-------------------------Brain of Robot Portion-------------------------------//

  if (rsensorState == LOW){ // if right front down sensor hits the white
    if (phase != 0){ //if phase is not set to 0
      phase = 0; //set phase to 0

    servo_R.write(0);
    servo_L.write(0);
    Serial.println("Right Sensor");
    delay (250);
    }
  }

  if (lsensorState == LOW){ //if left front down sensor hits the white
    if (phase != 1){ //if phase is not set to 1
      phase = 1; //set to 1

    servo_R.write(180); //spin robot accordingly to get back inbounds
    servo_L.write(180);
    Serial.println("Left Sensor");
    delay(250);
    }
  }
  if (lsensorState == LOW && rsensorState == LOW){
    servo_R.write(180);
    servo_L.write(0);
    delay(250);
  }
  if (bsensorState == LOW){ //if back down sensor hits the white
    servo_R.write(0);
    servo_L.write(180);
    Serial.println("Back Sensor");
    delay(250);
  }
  if (distanceleft > 20 && distanceright > 20 && rsensorState == HIGH && lsensorState == HIGH
&& bsensorState == HIGH){ //if robot is out of distance and our robot is completly inbounds
    if (phase == 0){
      servo_R.write(0); //scan spinning left
      servo_L.write(0);
      delay(250);
    }
    if (phase == 1){
      servo_R.write(180); //scan spinning right
      servo_L.write(180);
      delay(250);
    }
  }
  if (distanceleft < 20 && distanceright < 20 && rsensorState == HIGH && lsensorState == HIGH
&& bsensorState == HIGH){ //if the robot is within 20 inches and our robot is completly
inbounds
    servo_R.write(0); //straight forward
    servo_L.write(180);
    delay(250);
    }
```

```
  if (distanceleft < 20 && distanceright > 20 && rsensorState == HIGH && lsensorState == HIGH
&& bsensorState == HIGH){ //if the robot is within 20 inches of the left sensor but not the
right and our robot is completly inbounds
    servo_R.write(180); //spin left
    servo_L.write(180);
    delay(250);
  }

  if(distanceleft > 20 && distanceright < 20 && rsensorState == HIGH && lsensorState == HIGH
&& bsensorState == HIGH) { //if robot is within 20 inches of the right sensor but the not the
left and our robot is completly inbounds
    servo_R.write(0); //spin right
    servo_L.write(0);
    delay(250);
  }
} //End of loop

//-------------------Functions---------------------------//

//RETURNS DISTANCE FROM RIGHT DISTANCE SENSOR
float getDistanceright(int trigPinright, int echoPinright){
  float echoTimeright;                    //variable to store the time it takes for a ping to
bounce off an object
  float calculatedDistanceright;          //variable to store the distance calculated from the
echo time

  //send out an ultrasonic pulse that's 10ms long
  digitalWrite(trigPinright, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPinright, LOW);

  echoTimeright = pulseIn(echoPinright, HIGH);      //use the pulsein command to see how long
it takes for the
                                          //pulse to bounce back to the sensor

  calculatedDistanceright = echoTimeright / 148.0;  //calculate the distance of the object
that reflected the pulse (half the bounce time multiplied by the speed of sound)

  return calculatedDistanceright;               //send back the distance that was calculated
}

//RETURNS DISTANCE FROM LEFT DISTANCE SENSOR
float getDistanceleft(int trigPinleft, int echoPinleft){
  float echoTimeleft;                     //variable to store the time it takes for a ping to
bounce off an object
  float calculatedDistanceleft;           //variable to store the distance calculated from the
echo time

  //send out an ultrasonic pulse that's 10ms long
  digitalWrite(trigPinleft, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPinleft, LOW);

  echoTimeleft = pulseIn(echoPinleft, HIGH);      //use the pulsein command to see how long it
takes for the
                                          //pulse to bounce back to the sensor

  calculatedDistanceleft = echoTimeleft / 148.0;  //calculate the distance of the object that
reflected the pulse (half the bounce time multiplied by the speed of sound)

  return calculatedDistanceleft;               //send back the distance that was calculated
}
```