AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING

Mechatronics Engineering and Automation- Credit Hours Engineering Programs (CHEP)
Academic Year 2021/2022

UEL Module Code EG8422	**UEL Module Name** Machine Design	
ASU Course Code **MCT131**	**ASU Course Name** **Introduction to Mechatronics**	
	Semester **Fall 2021**	**Date of Submission**

AIN SHAMS UNIVERSITY

FACULTY OF ENGINEERING

International Credit Hours Engineering Programs (i.CHEP)

Intro to Mechatronics

Marble Track

By:

Ahmed Hisham Fathy Hassabou	19P4007
Mohamed Ahmed Mohamed Abdelhalim Mohanna	19P6218
Abdallah Fawzy Mohamed Mohamed Darwish	19P3122
Ehab Ashraf Makram Gomaa	19P5927
Mahmoud Ahmed Elsheikh	19P4907

Abstract

This document describes the process of developing a marble track from A to Z, as the final project of MCT131. Inspired from forests and African Savana, the electrical and mechanical mechanisms are integrated with the forests view. The track was built from four phases: lifting, color sorting, shooting, and Collection. The project contains a rotating disk which play the role of the lifting mechanism. Color sorting mechanism that directs the path of colored balls will take in the track. Two shooting mechanisms that is represented by catapult. Each shooting mechanism will shoot the balls at two funnels. The two funnels are connected by one track where all the balls are going to gather at one reservoir. The mechanical and electrical designs, constructions of the parts and other information are discussed in detail throughout the report.

Table of Content

List of Figures and Tables

1. Introduction

Mechatronics is a branch of engineering that focuses on integrating mechanical, electrical, electronic, software, telecommunication, and control systems for creating new designs, products and technologies that solves our daily life problems. One of the many projects that make a standing ground for students to integrate those systems is the marble track.

The Marble Track Project is one of the most famous projects that make a standing ground for engineering students, for upcoming projects that they make throughout the studying years. This project from a watcher point of view might be simple and maybe trivial for some people. However, when it comes to real work, this project contains many difficulties and many problems that students need to solve throughout the phases of the project. This project requires many skills to do it; at which students learn those skills throughout the project. It gives the students a view of what they are going to experience at their careers afterwards. In this project, students learn how to design and manufacture the parts of the project and to assemble them. Then integrate those manufactured designs with electrical systems, which is considered as one of the hardest phases of the project. The target is to make mechanical systems synchronize with electrical system which is the objective of Mechatronics.

Our Marble Track is a combination between the advanced mechanical/electrical designs with the beautiful nature of the forest and the African Savana. The motion of the ball represents the water flow of the river, describing its cycle from falling from the tops of the mountains to the bottom of the lakes. From this we tried to mix the mechanisms of the marble track with nature.

The project consists of four phases, each phase follows the other in a loop. The first phase is the lifting mechanism phase at which the ball moves from a low position to higher position. The second phase is the sorting phase. In this phase, the color of the ball determines the path the ball will take. Then we come with the third phase which is the shooting mechanism phase. After the ball takes its path, the ball is shoot at a target where it is taken to the fourth and the final phase. The final phase is where all the balls are gather and return to the first phase, where its states the process all over again.

2. System Requirements

2.1. Mechanisms

This marble track should contain three main mechanisms:-

- Shooting Mechanism: There should be one shooter that aims on two separate targets or two shooter each have their own target. Our choice was two shooter each aim to a different funnel. The shooting should occur in a projectile fashion.
- Sorting Mechanism: The sorting should be based on one category (ex: color, size, weight). Our choice was sorting based on color using a color sensor.
- Lifting Mechanism: A mechanism that lifts the marble from a low position to a high position. We chose a revolving wheel that takes the marble ball then lifts it by rotating where the ball leaves the wheel due to its inertia and the wheel's design.

2.2. Theme

The marble track should have a theme that is attractive to the kids. Our theme was the jungle as kids have always been fascinated by all kinds of animals. We will try to simulate a small sized jungle.

2.3. Goals

This design should be good example of a mechatronic system that shows a smooth integration between mechanical and electrical system. Moreover, the design should not only be attractive to the users but also safe.

`

3. Project Plan

3.1. Mood Board

First of all, a mood board was created in order to give us a preliminary idea on what we are going to do.

Figure 1 Mood Board

3.2. Mind Map

After being influenced by the mood board, a mind map was drawn in order to introduce to us as many ideas as possible. These ideas covered all aspects of our system, such as:-

- Mechanisms design
- Materials used
- Sensors types
- Actuators type

Figure 2 Mind Map

3.3. Tasks timeline

After finalizing both the mood board and the mind map the tasks timeline was set. The project was divided into 4 phases.

1. Software Design (Mechanical and Electrical Design): For the mechanical part, the inventor software was used while in the electrical part, the online platform tinker cad was used. The Duration of this phase was 2 weeks.

Figure 3 Tinker Cad logo

2. Mechanisms manufacturing, assembling and testing: Each part in all the mechanisms was manufactured. After the manufacturing processes the mechanisms was assembled, then the actuators were added and each mechanism was tested. The duration of this phase was 3 weeks
3. System integration: Each mechanism was fixed on the base and both the track and the sensors were added. The marble balls were added. The track was tested then all the encountered problems were added. The duration of this phase was 3 weeks.
4. Theme implementation: The decoration was bought and added. The duration of this phase was 1 week.

3.4. Sketches

Some sketches were made to discuss the variety of options that we had and give us an overview to how the track will look like.

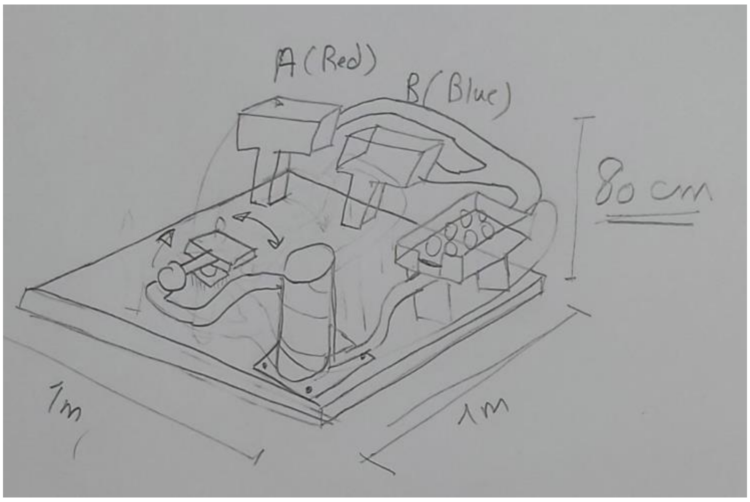

Figure 4 Track preliminary sketch

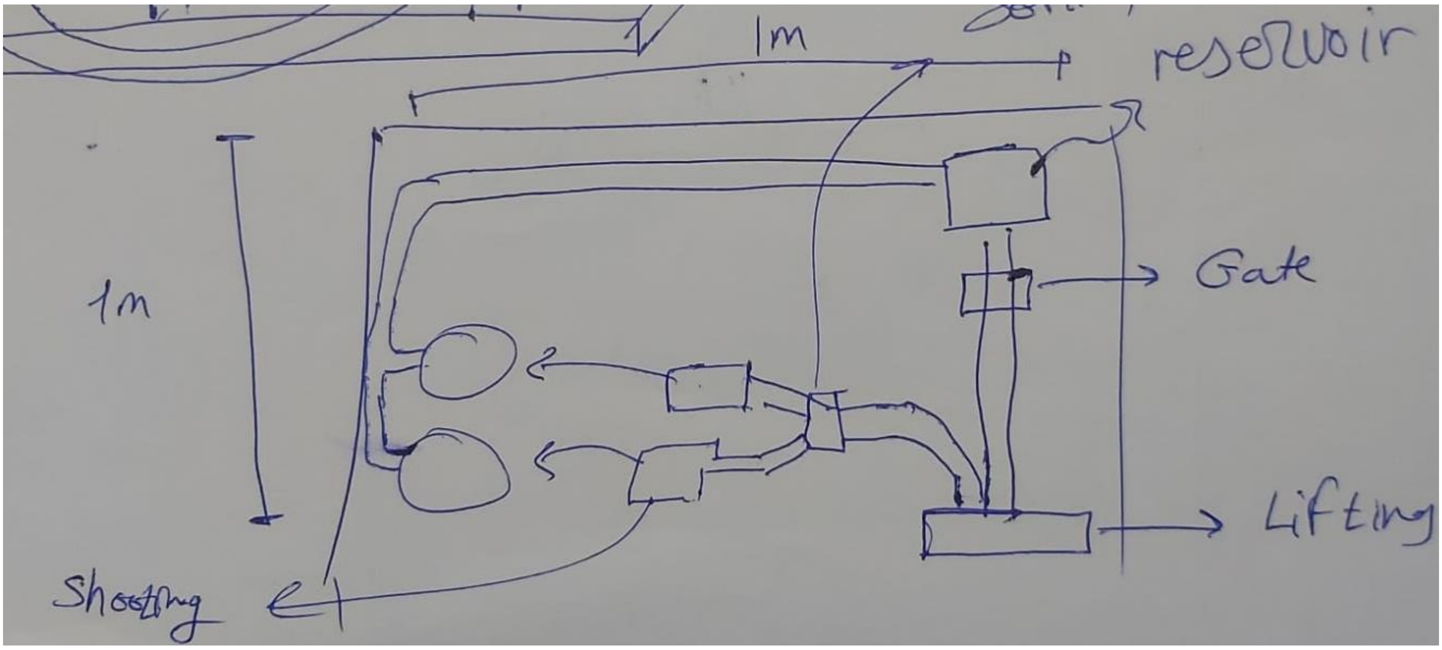

Figure 5 Track Final Sketch

5. Reservoir

5.1. Brief

The reservoir is the starting and finishing position of the track. In addition it is the place where the user can add new balls to the track.

5.2. Software Design

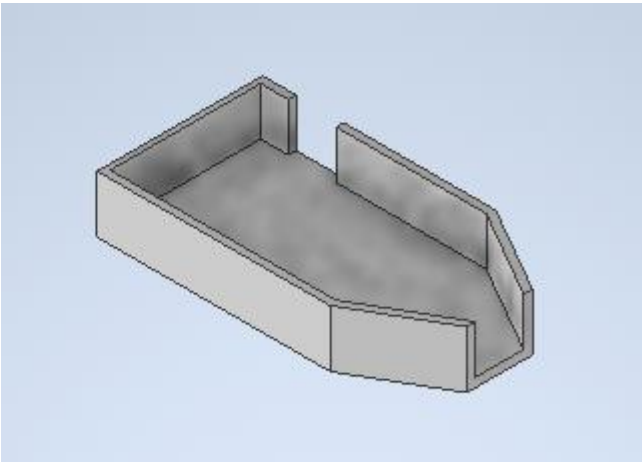

Figure 6 Reservoir Design

5.3. Manufacturing

The base of the reservoir was cut using laser cut while its sides were cut using an electrical hand saw as their details was not difficult. After cutting the reservoir parts, each side was glued to the base using glue that has a strong bonding force.

6. Gate

6.1. Brief

The Gate is the place where the user can control the flow of the balls. It can be done either manually as used in our track or automatically by attaching the gate to a motor via a four bar mechanism.

6.2. Design

The gate consists from two parts. A fixed part where the balls are fed to and a moving part that decides whether the ball is allowed to go to the track or not.

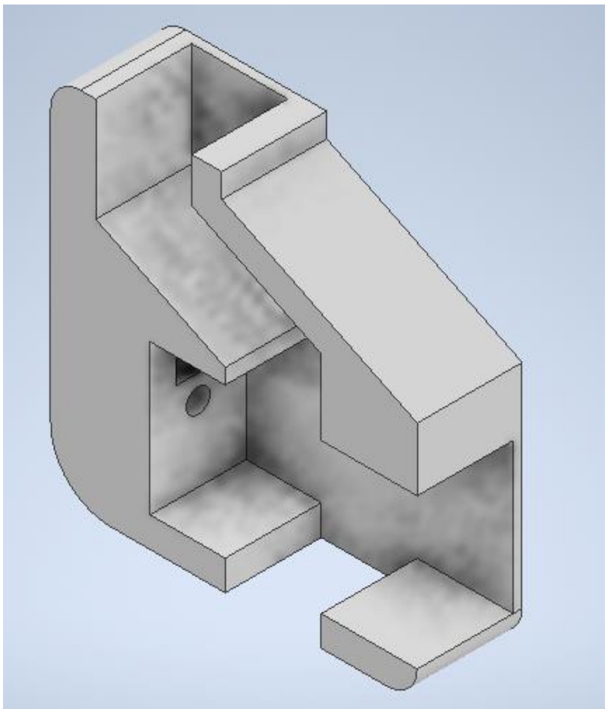

Figure 7 Gate's Fixed Part

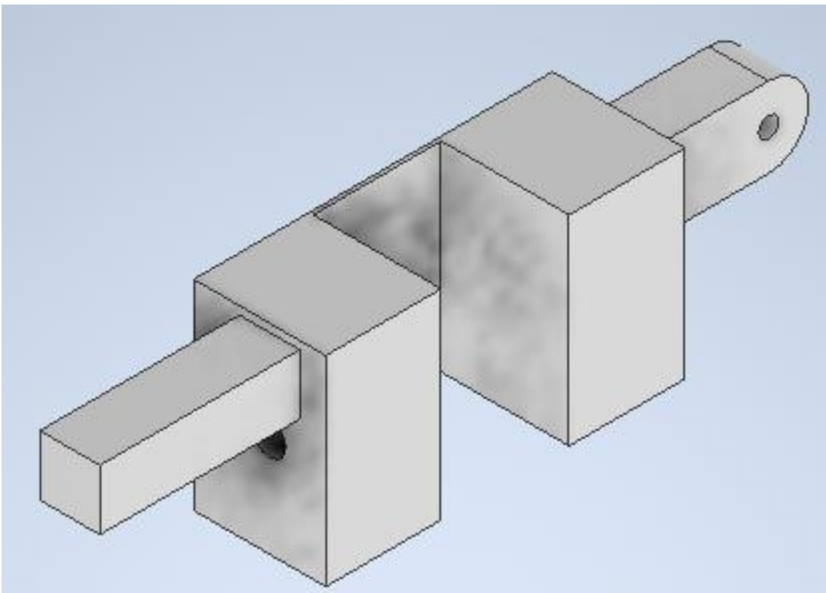

Figure 8 Gate's movable part

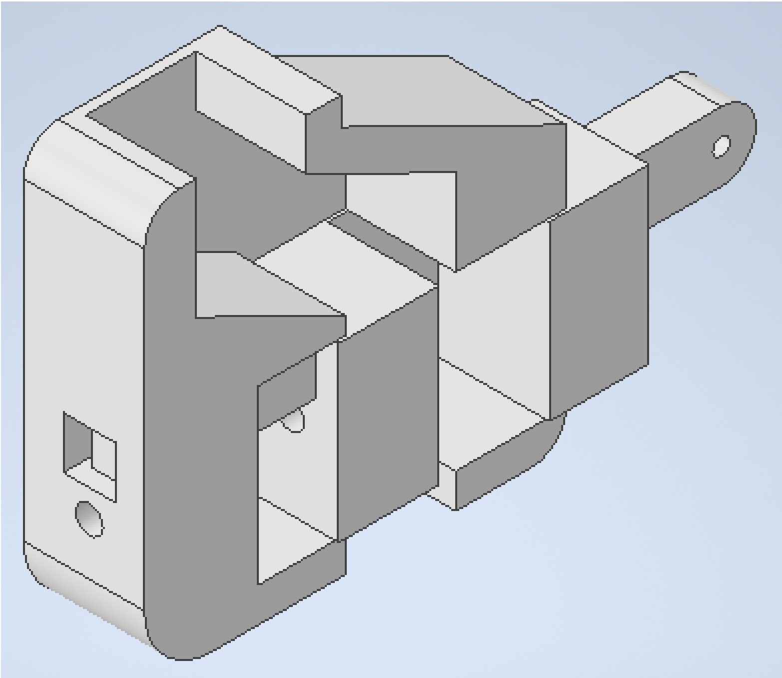

Figure 9 Gate's assembly

6.3. Manufacturing

Due to it complex design and big thickness it was manufactured using a 3d printer.

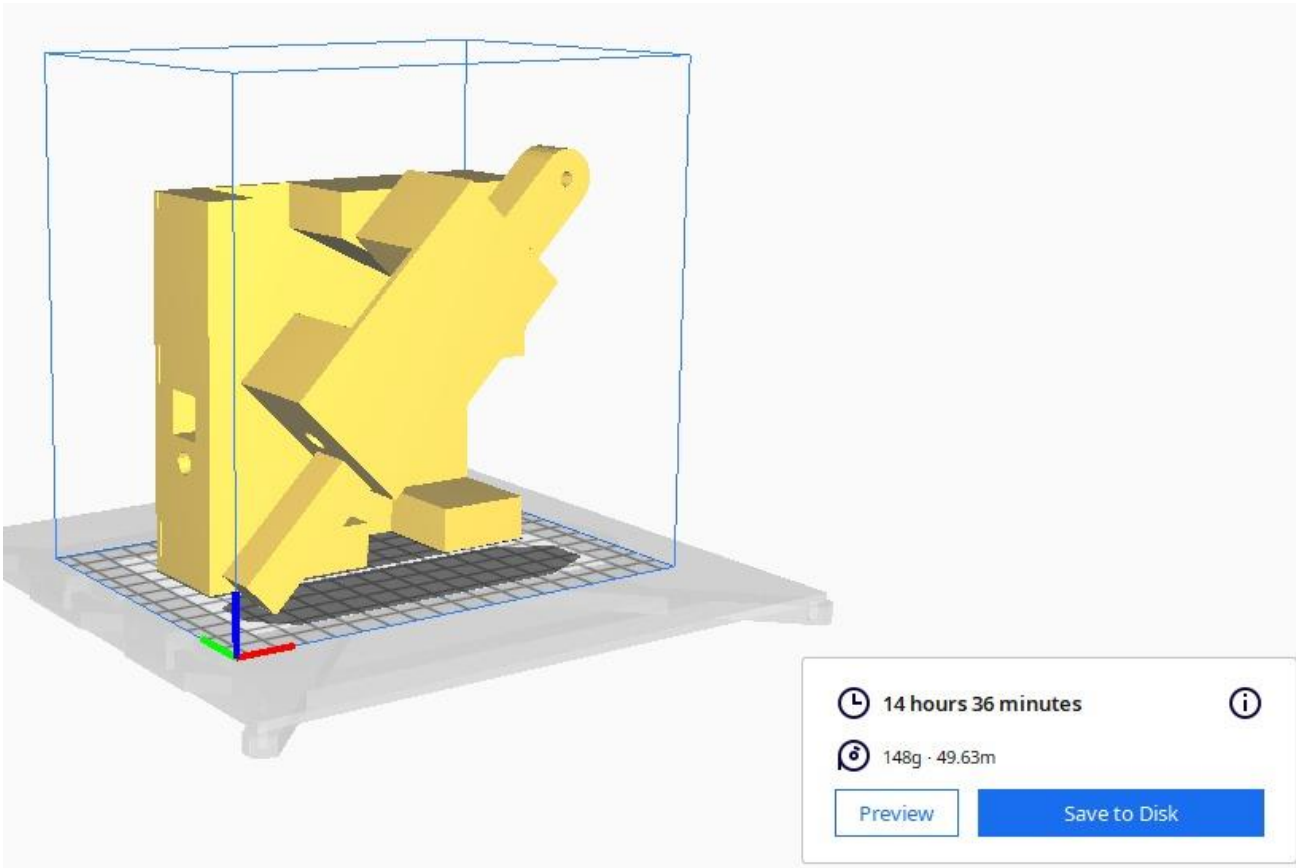

Figure 10 Gate on the 3d printer software

Figure 11 Movable Part

Figure 12 Fixed Part

7. Lifting Mechanism

7.1. Brief

The idea of the lifting mechanism is a revolving wheel that can hold the ball in any position expect one position at the top where the ball can leave the wheel using its own inertia. There is a big gear in the back of these wheel to transfer the angular motion from a smaller gear that is attached to a Dc motor

7.2. Design

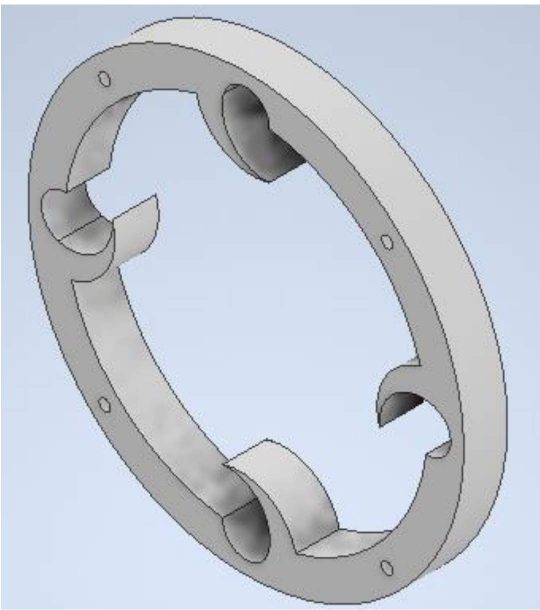

Figure 13 Lifting wheel design

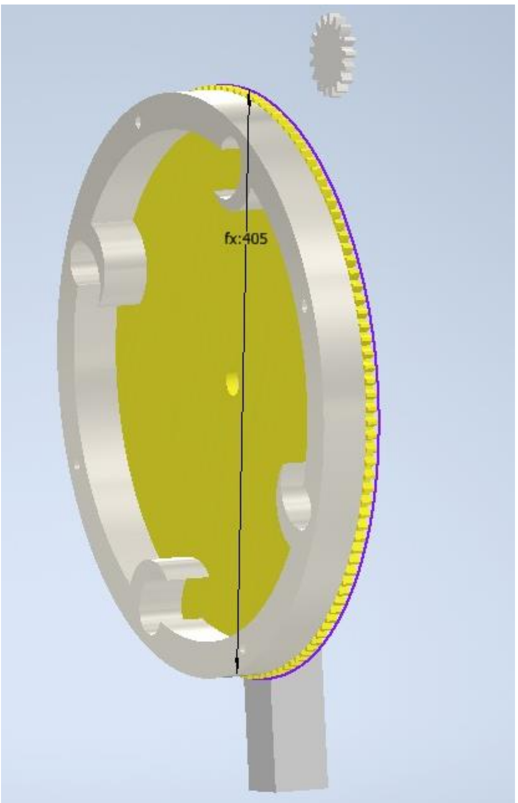

Figure 14 Lifting Assembly

7.3. Manufacturing

We used a laser cut to cut several layers of the revolving wheel then glued then together

Figure 15 revolving wheel layers

Figure 16 Revolving wheel

The gears were also cut using lase cut

Figure 17 Lifting mechanism gears

Later a bearing was put in the center of the gear to help fix the lifting on a fixed Shaft.

7.4. Motor Sizing

Figure 18 Dc Motor

- Rated Power: 12 V
- Max speed: 133 rpm
- Stall Torque : 20Kg.cm

We controlled this dc motor speed using H- Bridge connection and Arduino to make it fit our requirements.

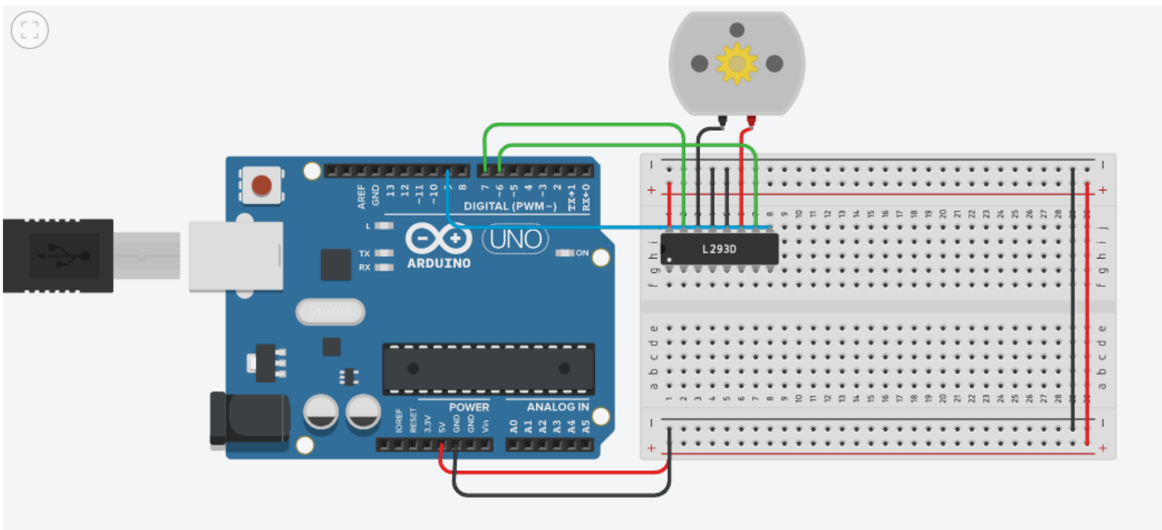

Figure 19 Dc Motor Tinker Cad Connection

Several values of speed was tested on this lifting to determine the best speed suited for the lifting. The used code to control and test the speed of the motor. Same connection that was used in tinker cad but the pin numbers may differ.

```
Lifting_Speed_Control §

#define enA 9
#define in1 6
#define in2 7
void setup() {
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  analogWrite(enA,45);
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
}
void loop(){
}
```

Figure 20 Lifting Speed Control Code

8. Shooting Mechanism

8.1. Brief

The idea of our shooting mechanism is a rotating cam that elevates the rod to a certain position then due to its design the rod position changes suddenly and with the aid of rubber bands the rod now have enough force to shoot the ball.

8.2. Design

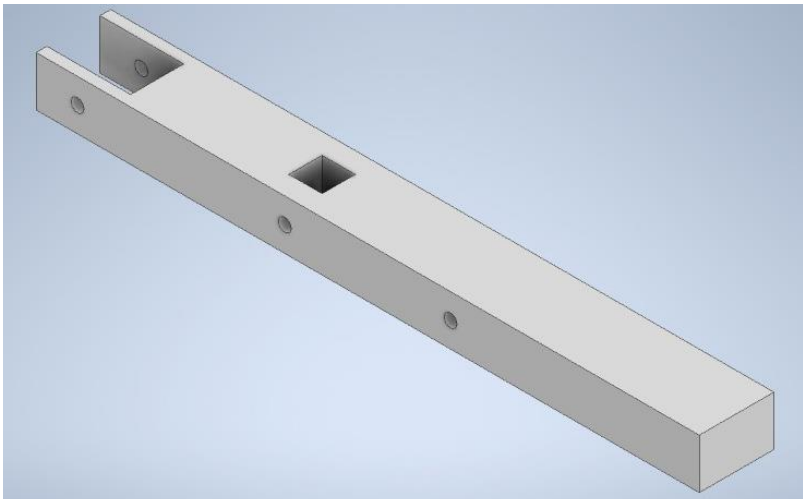

Figure 21 Rod

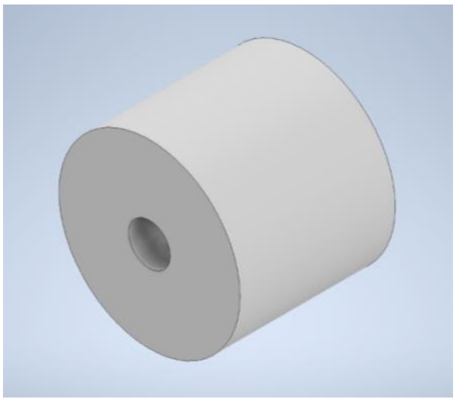

Figure 22 Roller

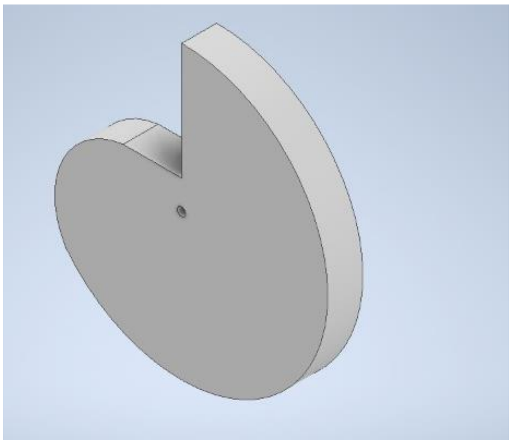

Figure 23 Cam

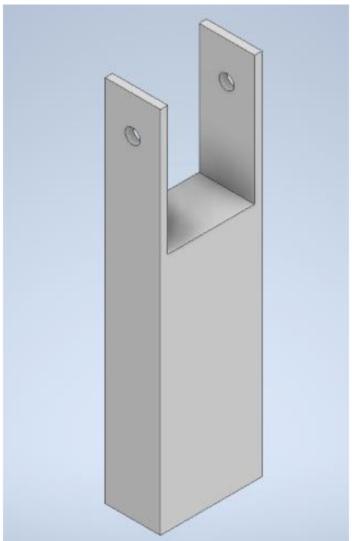

Figure 24 Rod Fixation

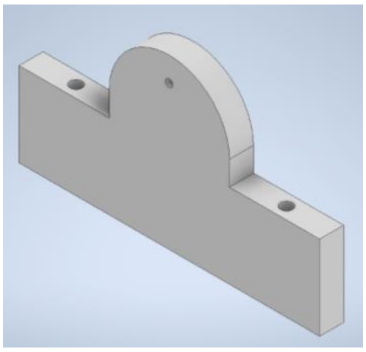

Figure 25 Cam Fixation

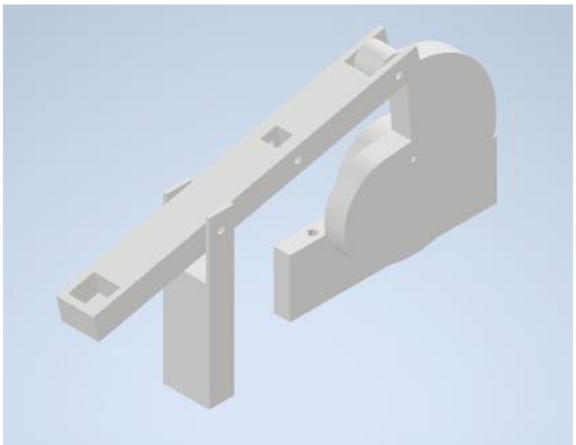

Figure 26 Full Assembly

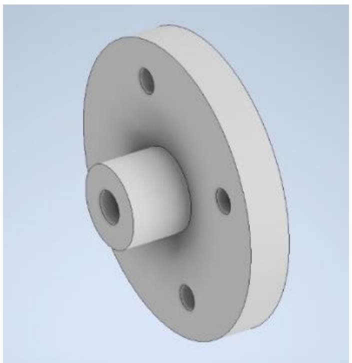

Figure 27 Spacer

Later on it was found that a gear and a spacer was needed to facilitate the transfer of motion from the motor to the cam

8.3. Manufacturing

The roller and spacer were manufactured via 3d printing while other parts were manufactured via laser cut as they were cut into layers then glued together

Figure 28 Spacer

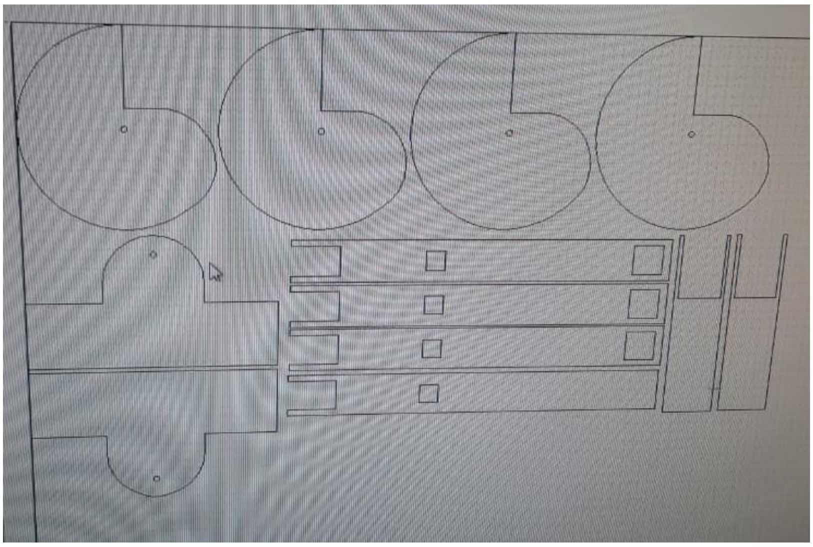

Figure 29 Parts on the laser cut machine software

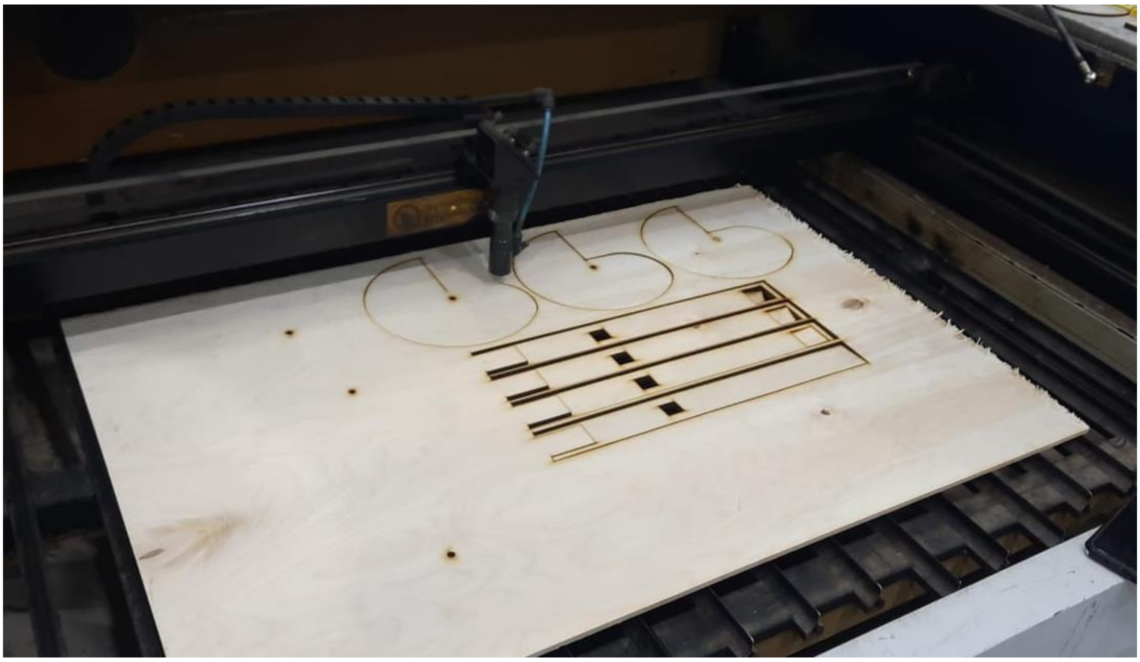

Figure 30 Process of Laser Cut

Figure 31 Final Parts

8.4. Assembly

Figure 32 Gear Cam Assembly

Figure 33 Full Assembly

8.6. Motor Sizing

The same motor that was used for the lifting was also used for the shooting but we did not only test the best speed but also we tested for the time that takes the cam to make a complete revolution “one shot”.

For shooting 1:-

Shooting_1_speed_control

```
#define enA 9
#define in1 6
#define in2 7
#define button 4
void setup() {
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
}
void loop(){
  analogWrite(enA,255);
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  delay(750);
   analogWrite(enA,0);
    digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  delay(15000);
}
```

Figure 34 Shooting 1 Speed Control

Shooting 2:-

Shooting_2_speed_control

```
#define enA 9
#define in1 6
#define in2 7
void setup() {
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
}
void loop(){
  analogWrite(enA,190);
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  delay(820);
  analogWrite(enA,0);
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  delay(15000);
}
```

Figure 35 Shooting 2 Speed Control

9. Sorting

9.1. Brief

Our sorting Mechanism is a gate mounted on the top of a servo motor that changes its position according to an the output of a color sensor

9.2. Design

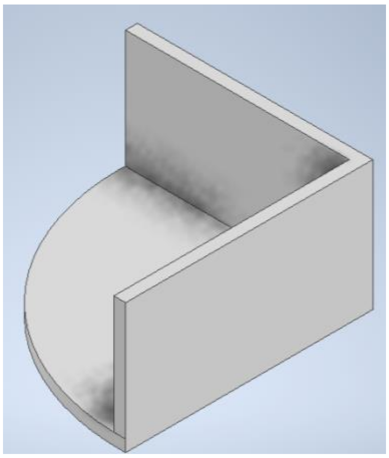

Figure 36 Sorting Gate

9.3. Manufacturing

It was made from strong card board as it is lightweight and will not be a problem to mount it on to of the servo motor.

9.4 Motor Sizing

- Operating Voltage 5V
- Torque 1.8 kg.cm

We tested the servo motor to determine its zero, 90 degrees and 180 degrees Positions.

Figure 37 Servo Motor

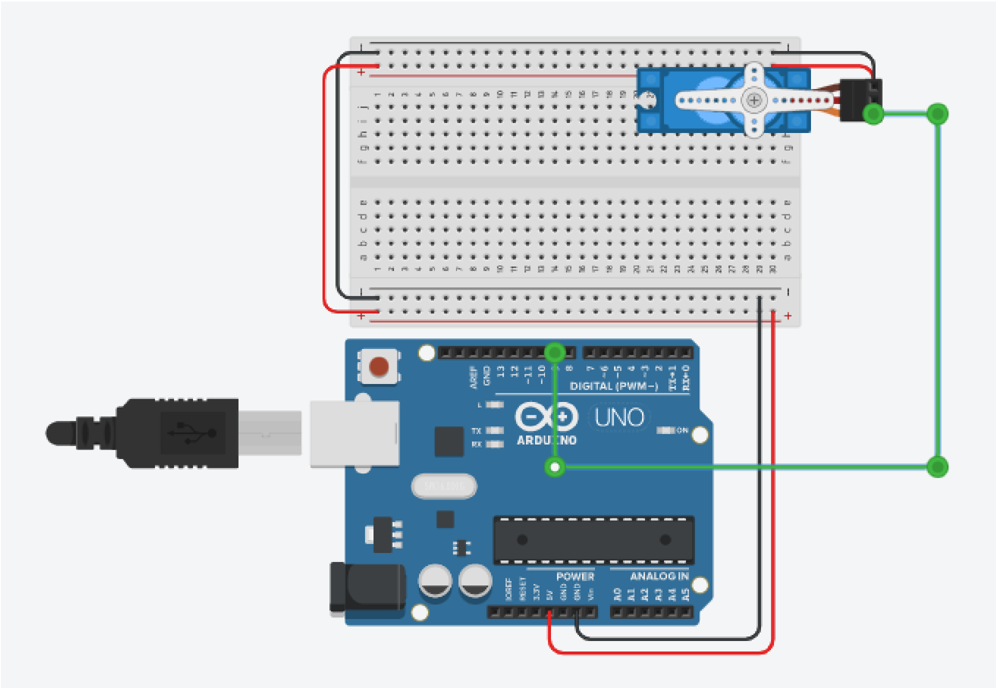

Figure 38 Servo Connection

Servo_Position_Test§

```
#include <Servo.h>
Servo myservo;
void setup() {
  myservo.attach(9);
  myservo.write(90);
}

void loop() {
}
```

Figure 39 Servo position Test code

10. Sensors

10.1. Overview

In our design we only used two sensors. The first one is the Color Sensor "TCS3200" With Focusable Lens and the second one is the motion Sensor

10.2. Color sensor

The reason behind choosing this color sensor was having its own lens causing it to be more accurate. This sensor have four arrays of 16 photodiodes. Each array represent a primary color while the last array is called the unfiltered array. These arrays gives the sensor 64 sensing point. The color sensor have 8 pins. Two pins four the Vcc and ground. A pin for the leds mounted on the sensor. Two pins "S_0,S_1" for the frequency scaling. This frequency scaling allows this sensor to be applicable with many micro controllers. The following picture explains the frequency scaling.

S0	S1	Frequency Scale
L	L	POWER DOWN
L	H	2%
H	L	20%
H	H	100%

Figure 40 TCS3200 Frequency Scaling

It was found that the best scaling for the Arduino was at 20%. S_2 and S_3 input pins for sensing they also have a combination to determine which color is being sensed.

S2	S3	Photodiode
L	L	RED
L	H	GREEN
H	L	CLEAR
H	H	BLUE

Figure 41 Sensing Array of color sensor

The last pin is the output pint that gives a feedback of the sensed colors.

Calibrating the color sensor to determine which values are equivalent to our ball colors was done using the following code

Calibration

```
#define S0 4
#define S1 5
#define S2 2
#define S3 3
#define sensorOut 8
#define led 12
int redPW = 0;
int greenPW = 0;
int bluePW = 0;
 void setup() {
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(sensorOut, INPUT);
  pinMode(led,OUTPUT);
  digitalWrite(led,1);
  digitalWrite(S0,HIGH);
  digitalWrite(S1,LOW);
  Serial.begin(9600);
}
 void loop() {

  redPW = getRedPW();
  delay(200);
  greenPW = getGreenPW();
  delay(200);
```

Figure 42 Calibration Code Part 1

```
  bluePW = getBluePW();
  Serial.print("Red PW = ");
  Serial.print(redPW);
  Serial.print(" - Green PW = ");
  Serial.print(greenPW);
  Serial.print(" - Blue PW = ");
  Serial.println(bluePW);
}
int getRedPW() {
  digitalWrite(S2,LOW);
  digitalWrite(S3,LOW);
  int PW;
  PW = pulseIn(sensorOut, LOW);
  return PW;
}
int getGreenPW() {
  digitalWrite(S2,HIGH);
  digitalWrite(S3,HIGH);
  int PW;
  PW = pulseIn(sensorOut, LOW);
  return PW;
}
int getBluePW() {
  digitalWrite(S2,LOW);
  digitalWrite(S3,HIGH);
  int PW;
  PW = pulseIn(sensorOut, LOW);
```

Figure 43 Calibration Code part 2

```
int getBluePW() {
  digitalWrite(S2,LOW);
  digitalWrite(S3,HIGH);
  int PW;
  PW = pulseIn(sensorOut, LOW);
  return PW;
}
```

Figure 44 Calibration Code Part 3

The Following Number represents the range for both the green and blue balls that we are going to use.

```
green ball: red(2000-11000) green (2000-4500) blue (3500-12000)

blue ball: red (5000-22000) green(2000-4500) blue(1500-3000)
```

Figure 45 Color Range

10.3. Motion Sensor

The motion Sensor gives an output value of 1 when something passes by. The motion sensor has three pins, two to Vcc and ground and one for the output. We tested this sensor to determine its range.

Figure 46 Motion Sensor

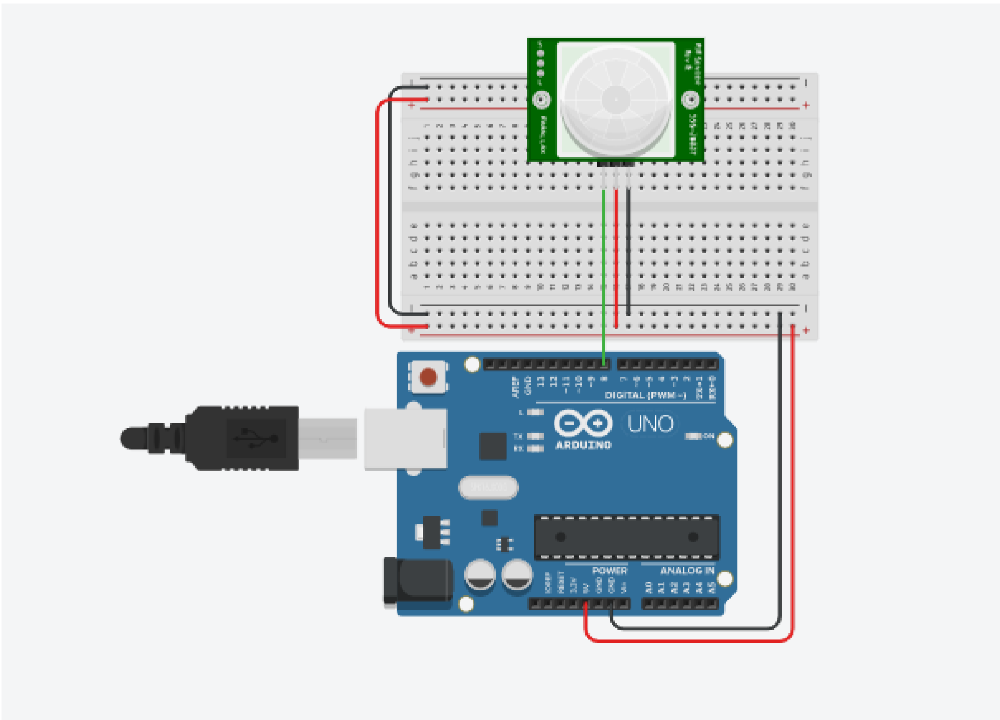

Figure 47 Motion Sensor Connection on Tinker Cad

Range_Test

```
#define motion 8
void setup() {
  pinMode(motion,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int out = digitalRead(motion);
  if(out == 1)
  Serial.println(out);

}
```

Figure 48 motion Sensor Range test

11. Electrical Design

11.1. Lifting

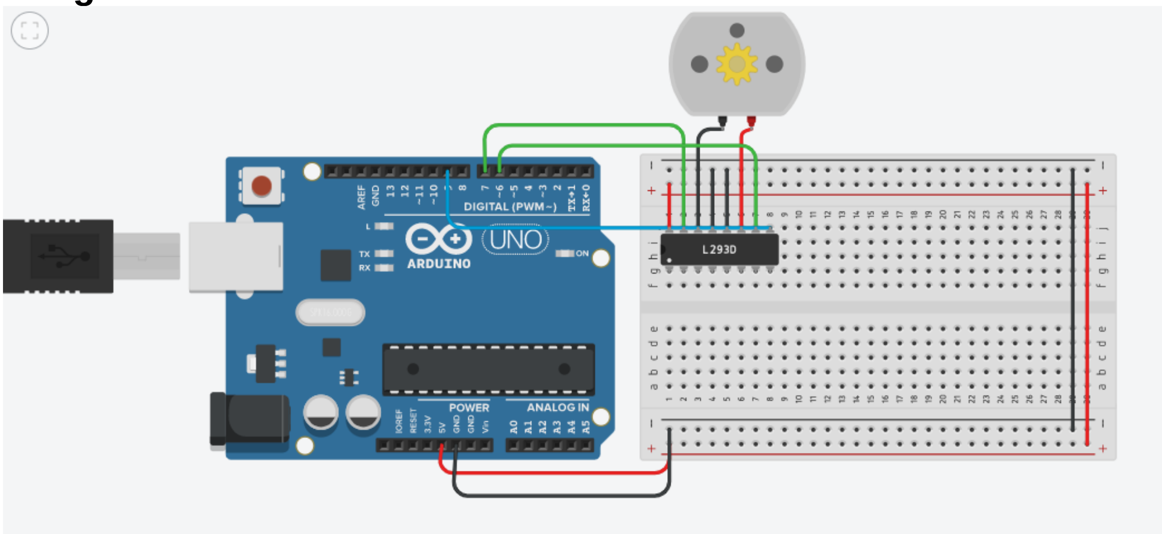

Figure 49 Lifting Electrical Design

Lifting §

```
#define enA 9
#define in1 6
#define in2 7
#define button 4
void setup() {
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(button, INPUT);
}
void loop(){

  analogWrite(enA,45);
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  delay(3500);
  analogWrite(enA,0);
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  delay(3500);
}
```

Figure 50 Lifting Code

11.2. Sorting and Shooting 1

There was no color sensor on tinker cad so we simulated using a motion sensor instead.

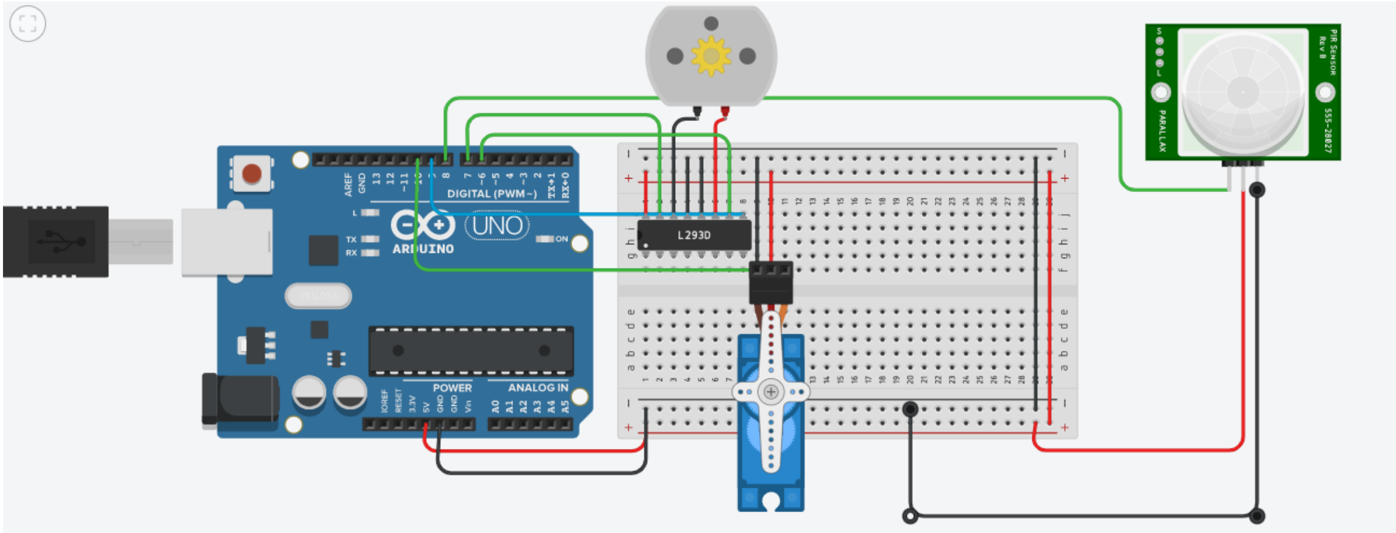

Figure 51 sorting-shooting 1 Connection

Soting_Shooting_1 §

```
#define S0 4
#define S1 5
#define S2 2
#define S3 3
#define sensorOut 8
#define led 12
#include <Servo.h>
#define enA 9
#define in1 6
#define in2 7
Servo myservo;
int redPW = 0;
int greenPW = 0;
int bluePW = 0;
 void setup() {
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(sensorOut, INPUT);
  pinMode(led,OUTPUT);
  digitalWrite(led,1);
  digitalWrite(S0,HIGH);
  digitalWrite(S1,LOW);
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
```

Figure 52 Sorting Shooting 1 Code Part 1

Soting_Shooting_1 §

```
  pinMode(in2, OUTPUT);
  analogWrite(enA,0);
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  myservo.attach(9);
  myservo.write(90);
}
 void loop() {
  redPW = getRedPW();
  delay(400);
  greenPW = getGreenPW();
  delay(400);
  bluePW = getBluePW();
  if(redPW <= 11000 && redPW >= 2000 && greenPW <= 4500 && greenPW >= 20000 && bluePW <= 12000 && bluePW >= 3500)
  {
    myservo.write(180);
    delay(1500);
    myservo.write(90);
    delay(1000);
    analogWrite(enA,255);
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    delay(750);
    analogWrite(enA,0);
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    }
```

Figure 53 Sorting Shooting 1 Code Part 2

Soting_Shooting_1 §

```
    if(redPW <= 22000 && redPW >= 5000 && greenPW <= 4500 && greenPW >= 2000&& bluePW <= 3000 && bluePW >= 1500)
    {
    myservo.write(0);
    delay(2500);
    myservo.write(90);
    }
}
int getRedPW() {
  digitalWrite(S2,LOW);
  digitalWrite(S3,LOW);
  int PW;
  PW = pulseIn(sensorOut, LOW);
  return PW;
}
int getGreenPW() {
  digitalWrite(S2,HIGH);
  digitalWrite(S3,HIGH);
  int PW;
  PW = pulseIn(sensorOut, LOW);
  return PW;
}
int getBluePW() {
  digitalWrite(S2,LOW);
  digitalWrite(S3,HIGH);
  int PW;
  PW = pulseIn(sensorOut, LOW);
  return PW;
```

Figure 54 Sorting Shooting 1 Code Part3

11.3. Shooting 2

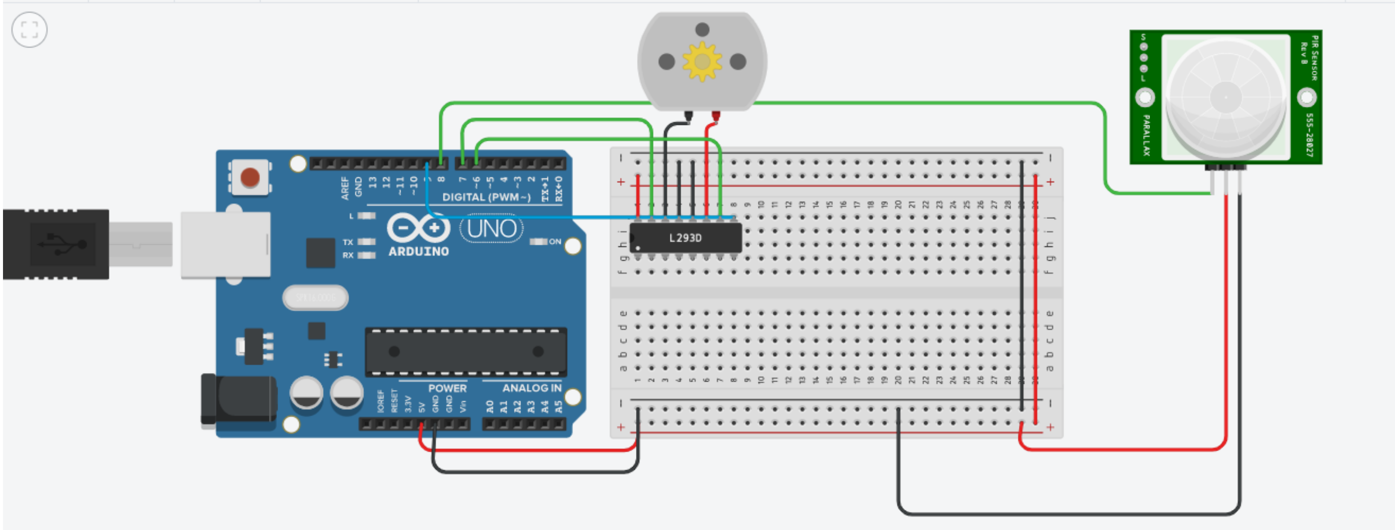

Figure 55 Shooting 2 Connection

Shooting_2

```
#define enA 9
#define in1 6
#define in2 7
#define motion 8
void setup() {
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
}
void loop(){
  int out = digitalRead(motion);
  if (out == 1)
  {
  delay(4000);
  analogWrite(enA,190);
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  delay(820);
  analogWrite(enA,0);
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  }
}
```

Figure 56 Shooting 2 Code

12. Flowchart

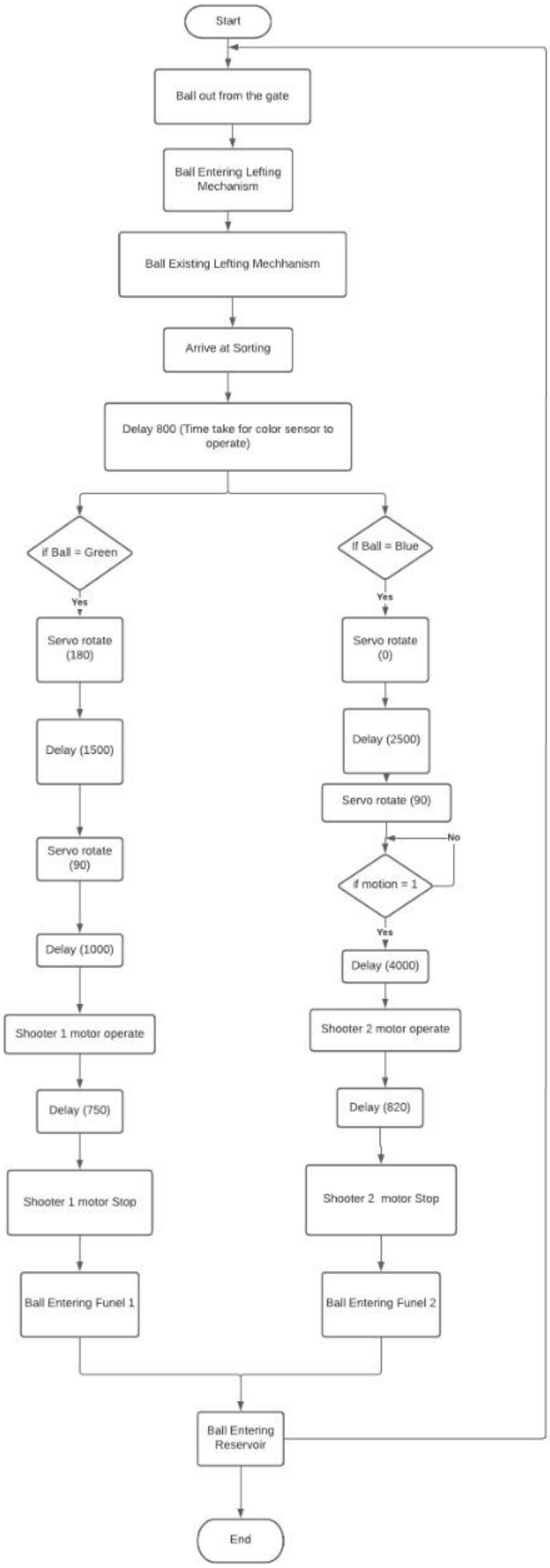

Figure 57 Flowchart

13. Theme Implementation

As mentioned before our theme was the jungle so, we purchased some artificial grass and anomal figures to help with the decoration.

Figure 58 Final Look of the marble track

14. Integration (Track overview)

The track leaves the reservoir and wait in the gate until the user open it up then the ball fall to the lifting mechanism were it rises to the track holding the sorting gate if the ball is green it falls to the first shooting and if the ball is blue the ball falls in the second shooter. Then the shooter shoots the ball into the funnel where it can go back to the reservoir and the loop is repeated.

15. Bill of Material

Table 1 Bill of Material

Item/Process	Quantity	Price per one/EGP	Total Price/EGP
Shooting 1&2 Laser Cut	-	300	300
Lifting Wheel Laser Cut	-	112	112
Lifting Gears Laser Cut	-	50	50
Bearing	1	5	5
Shaft 8mm	1	85	85
Reservoir Base laser Cut	-	30	30
Gate 3d Print	-	150	150
Shooter Roller 3D Print	2	10	20
Gear Spacer 3D Print	2	15	30
DC Motor	3	195	585
Servo Motor	1	120	120
Wooden Base	1	100	100
MDF Wood Squares	4	20	80
Strong Cardboard	5	10	50
Wood Nails	-	20	20
Washer & Nuts	-	20	20
Arduino Boards	3	120	360
Jumpers	-	40	40
Wood Slap	1	40	40
Artificial Grass	-	120	120
Animal Figures	-	75	75
Color Spray	3	20	60
Motion Sensor	1	40	40
Color Sensor	1	190	190
Female adaptors	5	5	25
H-bridge	3	60	180
Total			**2887**

16. Contributions

16.1. Ahmed Hisham Fathy Hassabou 19P4007

1. In Charge of Designing the Shooting Mechanism
2. Helped In Designing Lifting Mechanism, reservoir and Gate
3. Gate Fixation
4. Sorting Mechanism Fixation
5. Track & Funnels Manufacturing
6. Code Reviewing
7. Track Fixation
8. Motor speed Testing

16.2. Mohamed Ahmed Mohamed Abdelhalim Mohanna 19P6218

1. In charge of Designing reservoir and Gate
2. Helped In Designing Lifting and shooting Mechanism
3. Electrical Software Design
4. Searched For color sensor calibration Code
5. Writing Code (Test Codes & Processes Code)
6. Reservoir Fixation
7. Theme Implementation
8. Funnels Manufacturing
9. Report writing and organization

16.3. Mahmoud Ahmed Elshiekh 19P4907

1. In Charge of Designing the Lifting Mechanism
2. Helped In Designing shooting Mechanism, reservoir and Gate
3. Shooting 1&2 Assembly
4. Lifting Assembly
5. Lifting Fixation
6. Shooting Fixation
7. Track Fixation
8. Funnel Fixation
9. Sketches

16.4. Abdallah Fawzy Mohmed Mohamed Darwish 19P3122

1. Reservoir Fixation
2. Gate Fixation
3. In Charge of Designing the track
4. Track Manufacturing
5. Track Fixation
6. Color Sensor Calibration
7. Motor speed testing

8. Theme Implementation
9. Electrical Connections

16.5. Ehab Ashraf Makram Gomaa

1. In charge of designing Sorting Gate
2. Shooting Mechanism Fixation
3. Lifting Mechanism Fixation’
4. Servo Position Test
5. Mind Map
6. Mood Board
7. System Flowchart
8. Theme Implementation
9. Video Editing
10. Wrote the introduction and Abstract