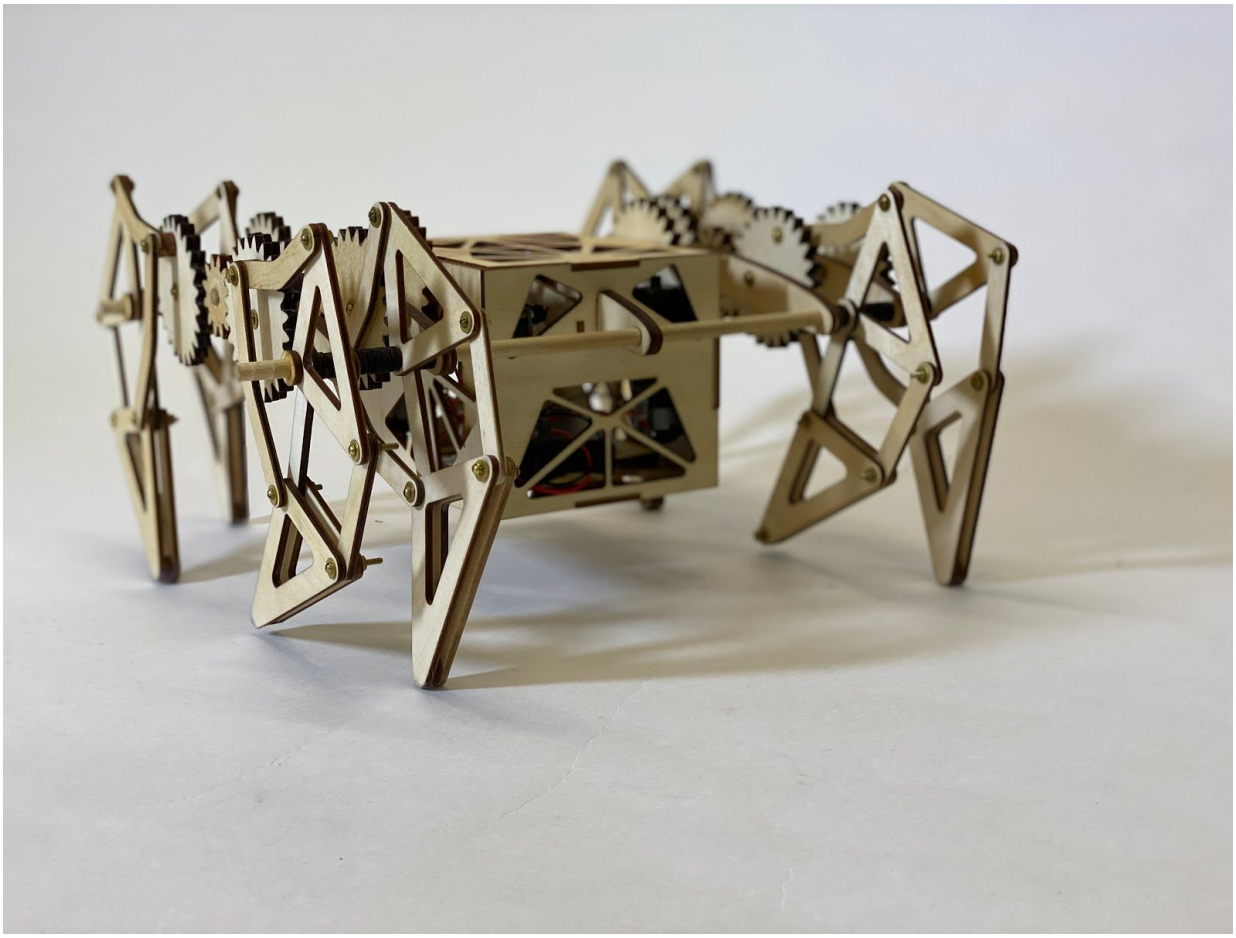
Lady Strandbeest



Created by **Ritik Batra** & **Tina Taleb**

DES INV 22
December 11, 2019

Table of Contents

Final Product Photo

Video Link

Bill of Materials

Description

Product Photos

Process Photos

Process Description

Reflection

Files

Bibliography

Video Link

Here is the YouTube video for our bot.

Bill of Materials

Item	Description	Qty
Wood	Used ⅛ inch depth wood pieces for laser cutting the wheels	10 sheets
Adafruit Feather 32u4 BLE	Used to control the car and receive commands through Bluetooth	1 pc
Battery Pack	Used to power the motors with 4 AA batteries	1 pc
LiPo Battery	Used to power the feather module	1 pc
Motors	Used to move the wheels	2 pc
Dowel	Used for attaching the legs together and keeping it aligned	3 pc
Screws & nuts	Used for connecting joints and keeping it sturdy	32 sets

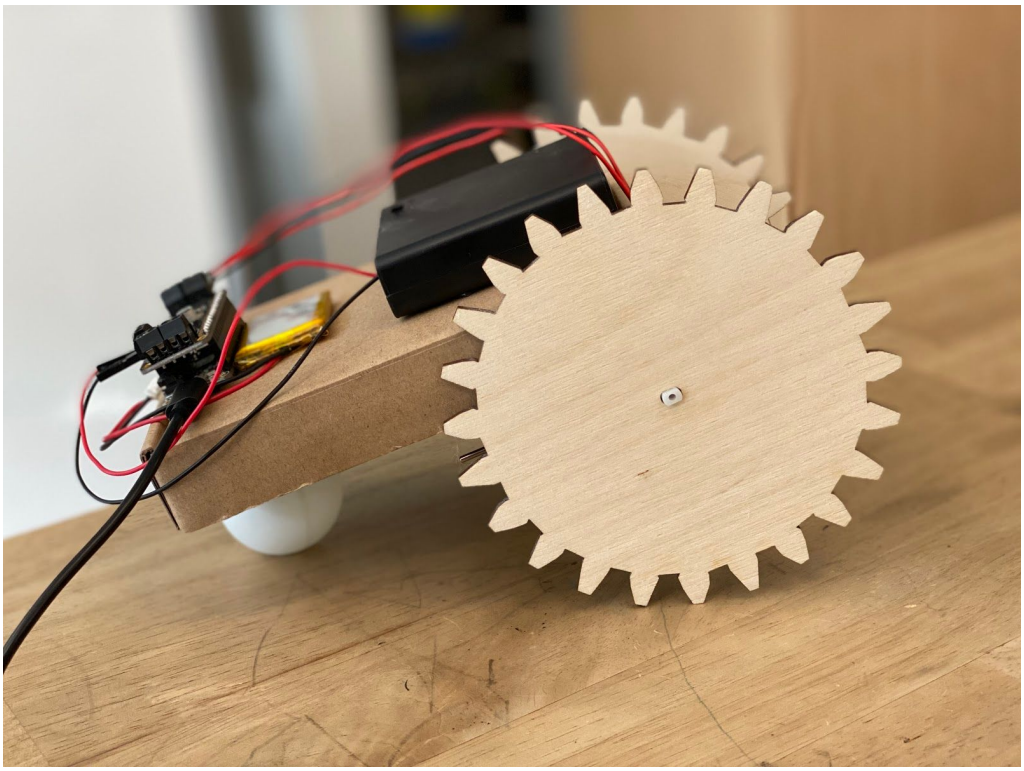
Tool	Info
Laser Cutter	Used to cut wood
Wood Glue	Used as a secondary adhesive to attach small pieces together
Super Glue	Used as a tertiary adhesive to make more permanent, quick connections
Plasti-Dip	Used to add friction to the bottom of the legs
Beeswax	Used to make the gear movement smoother

Description

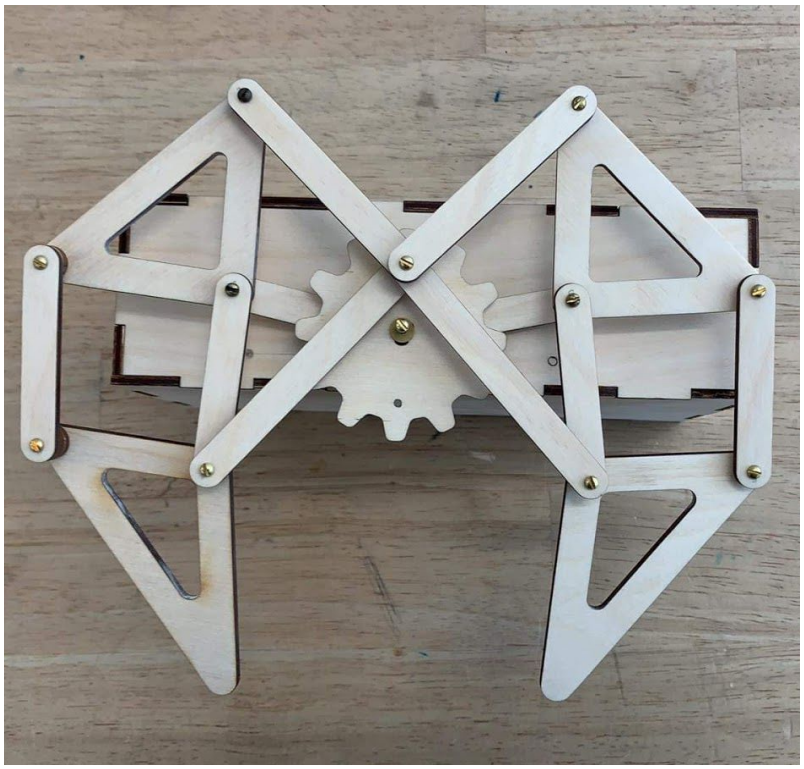
The Lady Strandbeest is an elegant bot that walks on her 8 legs in unison to overcome obstacles and maneuver to the endpoint. Using gear movement and precise measurements inspired by the famous Strandbeest by Theo Jansen, we designed and fabricated our Lady Strandbeest that is 16 times smaller than the original model. However, our Lady Strandbeest is fully motorized unlike the original model that operated by using wind and mechanical muscle system. With two motors attached to each side of the Lady Strandbeest, one gear dictates each side which required careful alignment to ensure smooth movement forward and turning.

Product Photos

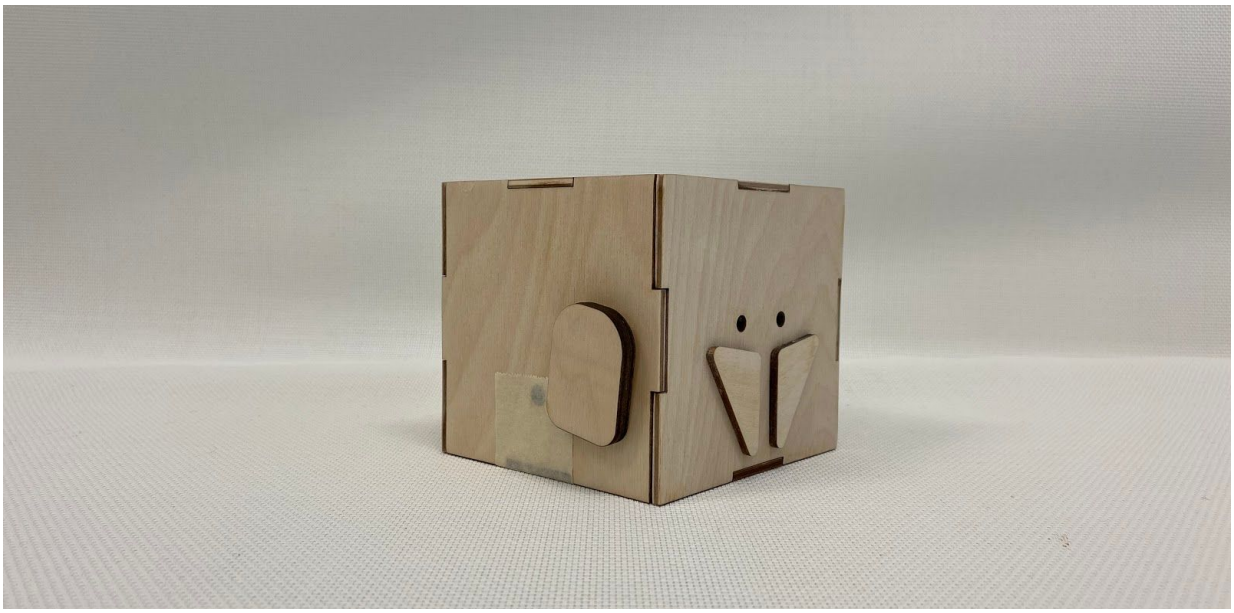
Week 1



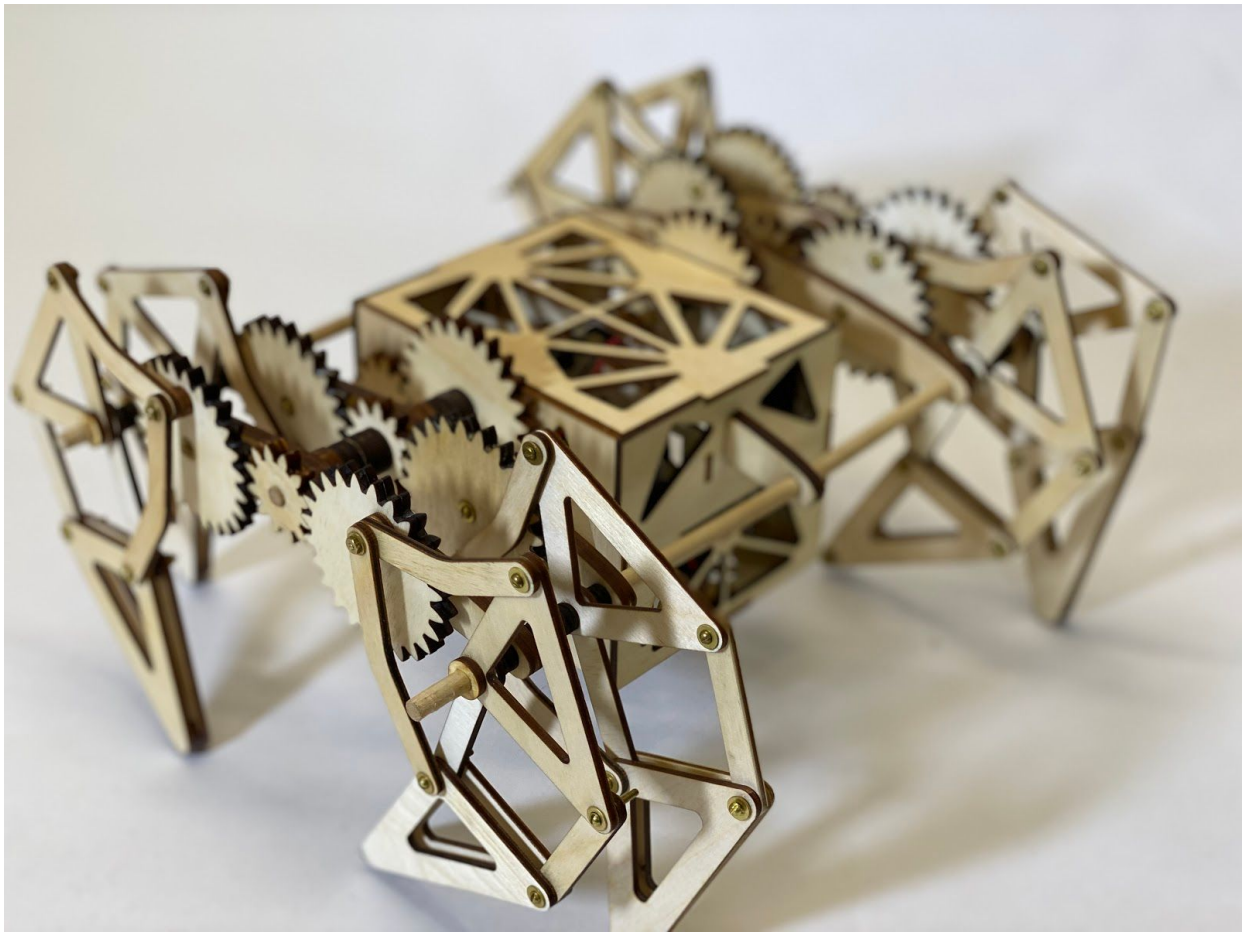
Week 2



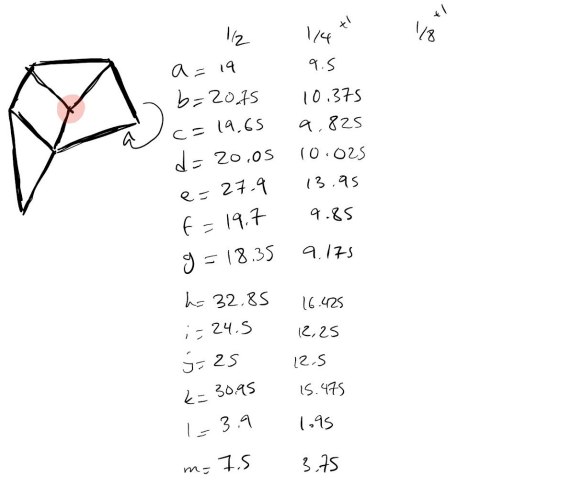
Week 3

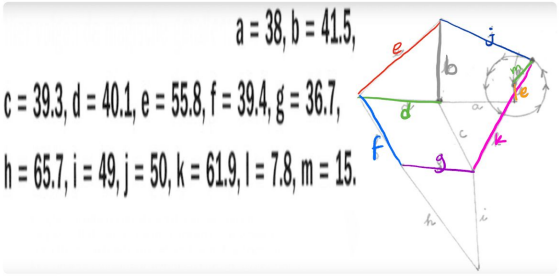


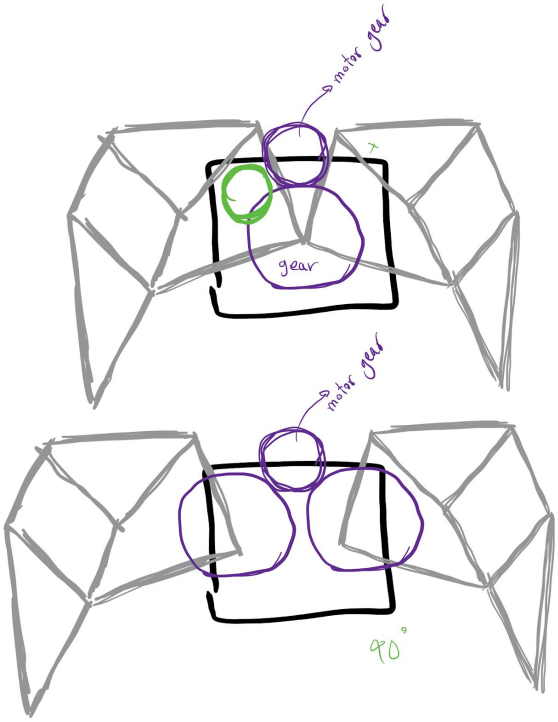
Week 4

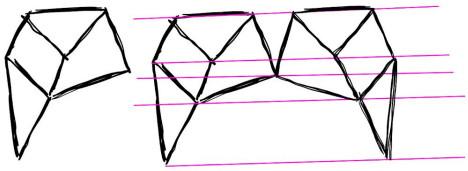


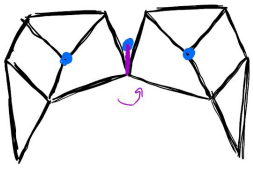
Process Photos

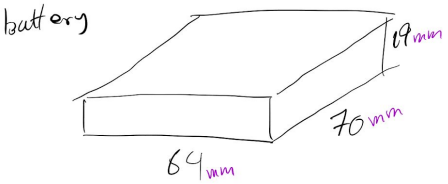


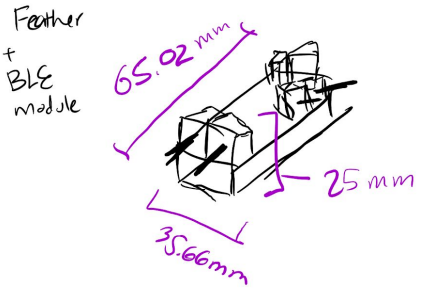


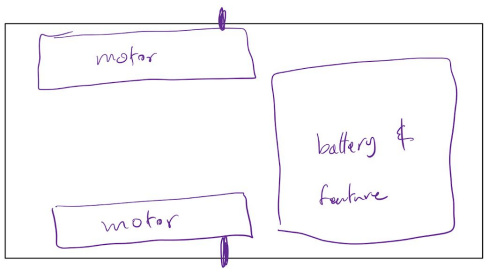




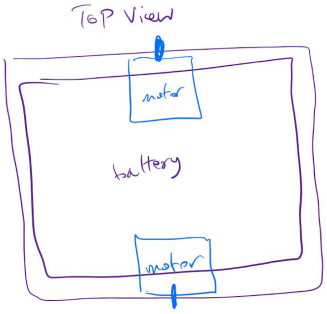


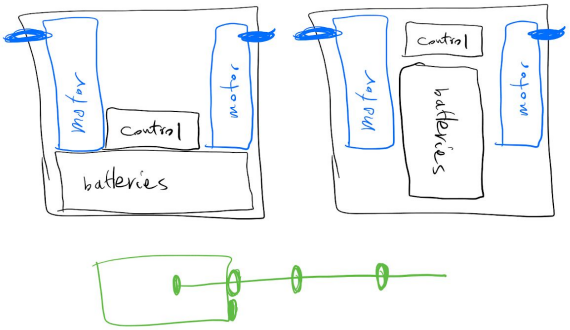






6.4mm rod for gears



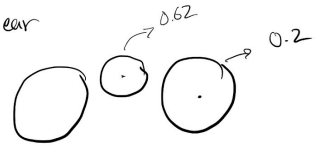


PARTS

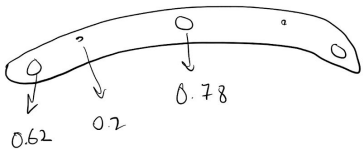
GEARS

big, 0.2,02	x 8	
big, 0.62	x 2	
small motor	x 2	
Small, 0.62	x 4	
PANELS	x 4	
DOWELS	x 4	
WASHERS		

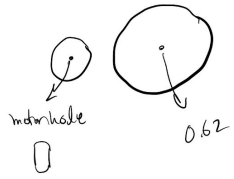
leg gear (x4)



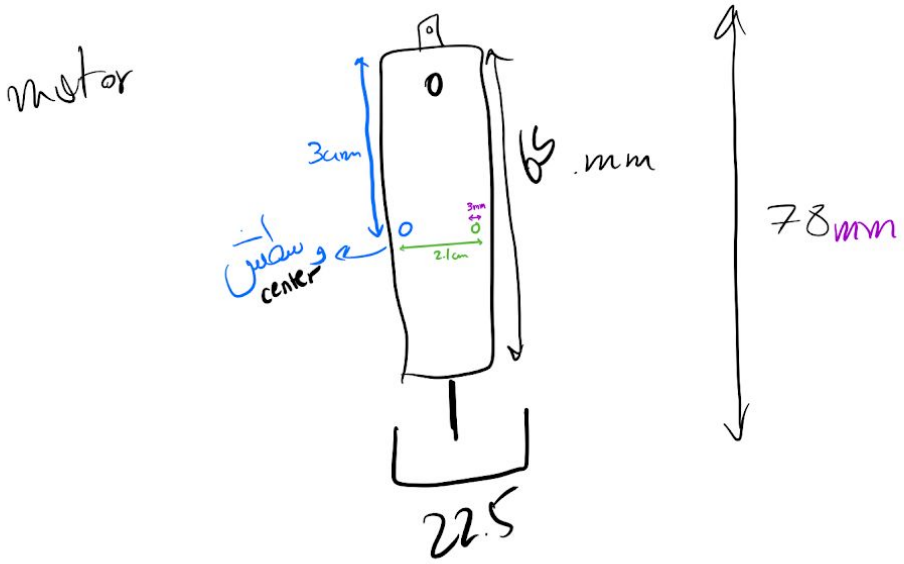
Panel (x4)

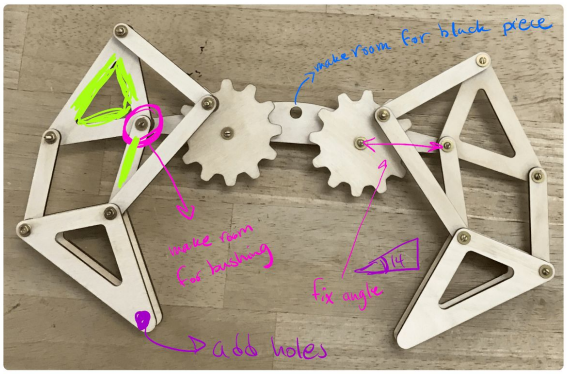


Motor gears (x2)



big, 0.2,02	x 8	8
big, 0.62	x 2	4
small motor	x 2	4
Small, 0.62	x 4	8
PANEL	x 4	8
triangle	x 8	
C	x 8	
big triangle	x 6	





big washers →

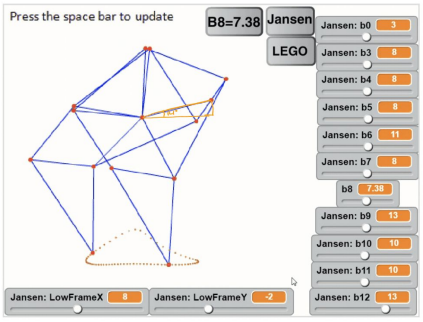
Gears
- ratio 2:1 for motor
- same for legs

big 51mm
small

bushing → (leg x3) x4

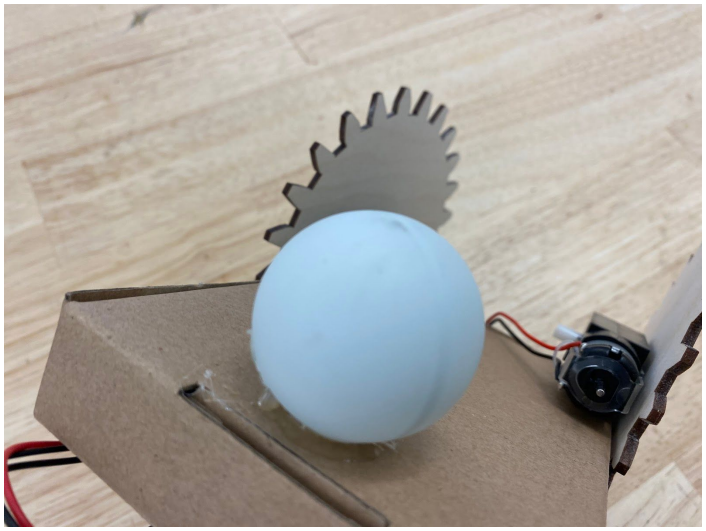
box x4

16

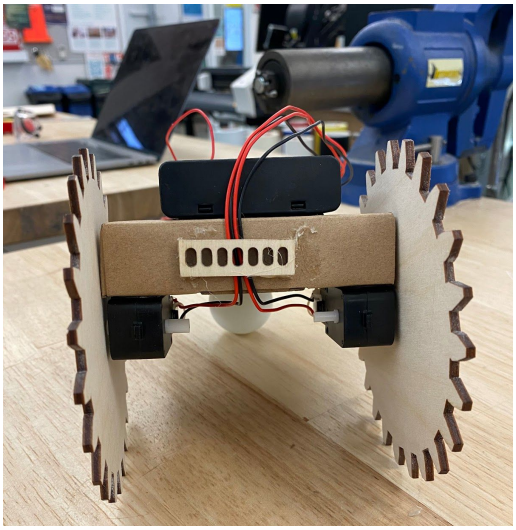


Strandbeest Optimized for LEGO

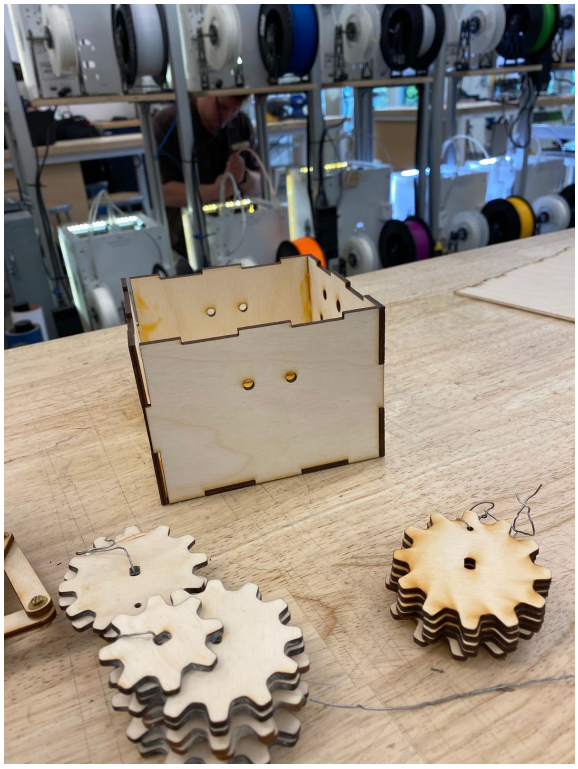
A



B

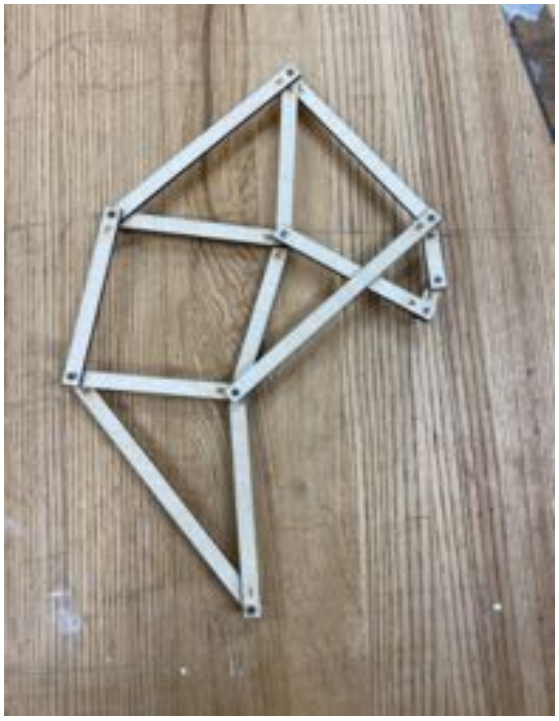


C



D

E

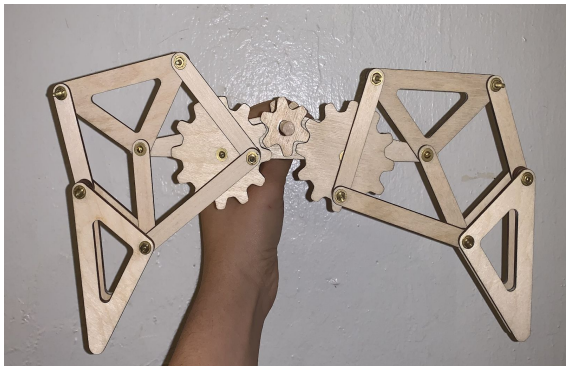


F

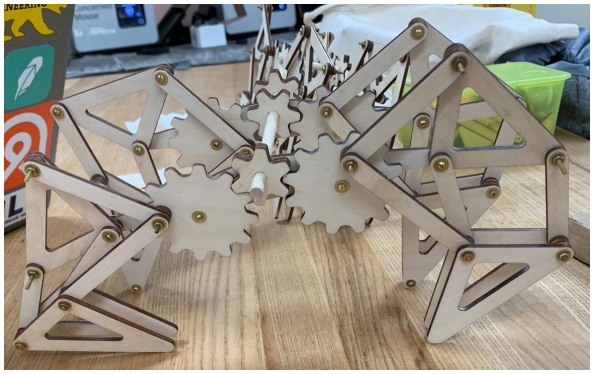
G

H

I



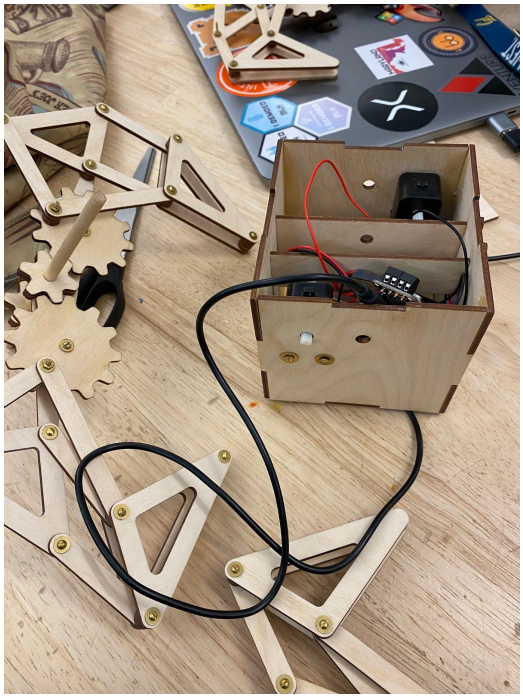
J



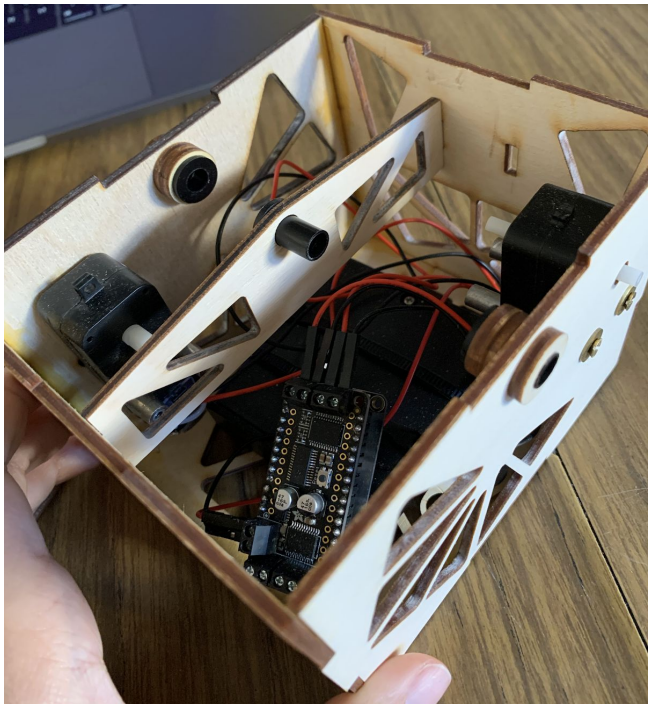
K

L

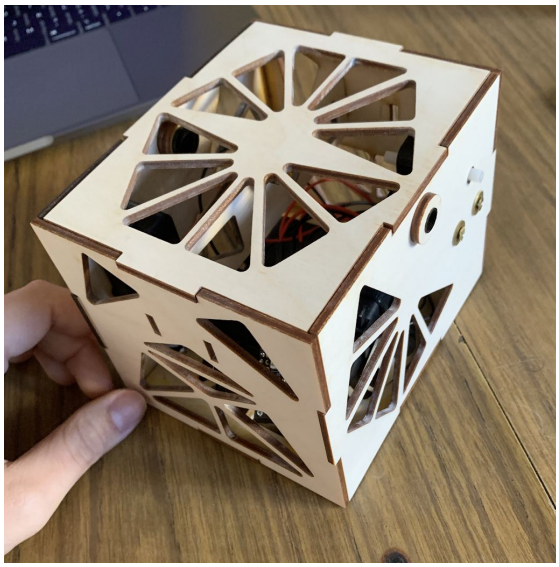
M



N

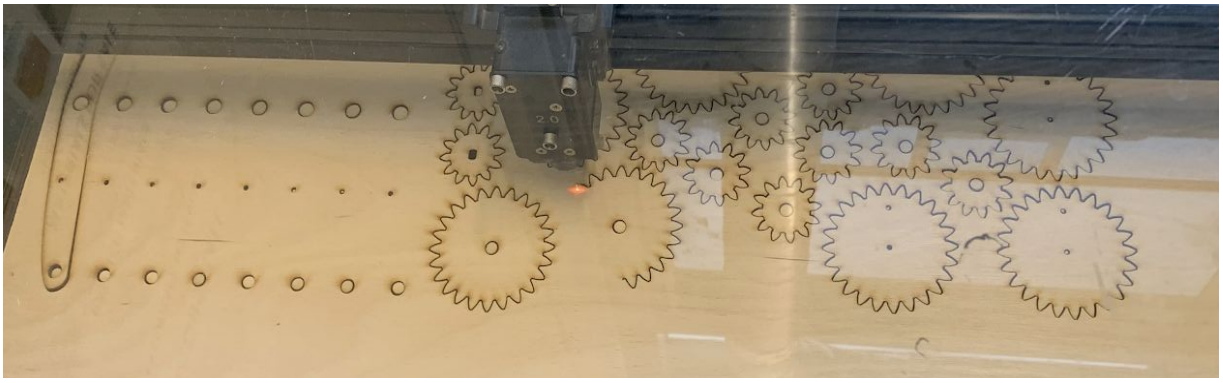


O



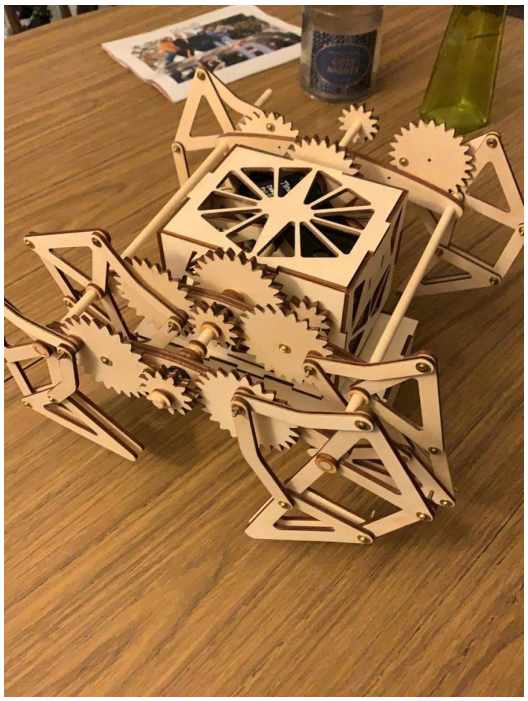
P

Q



R

S



T

A: Drawings and designs used for measurements and ideating [Assorted]
B: Hot Glued Ping Pong Ball for balance and stability [Week 1]
C: Used a test cut piece to keep wires more organized [Week 1]
D: Main body box (2) [Week 2]
E: Sanded and assembled box and legs [Week 2]
F: First assembled leg to examine functionality [Week 1]
G: First leg pieces [Week 1]
H: First assembled leg with gear [Week 2]
I: Discovering correct layering order [Week 2]
J: First assembled set of legs with gears and panel [Week 2]
K: 4 legs connected by dowels and gears [Week 2]
L: Laser cutting pieces for first iteration [Week 1]
M: Gluing first body [Week 1]
N: Cubic body with electronics and set of legs [Week 3]
O: New body with cut-outs and dowel stabilizers (Interior) [Week 4]
P: New body with cut-outs and dowel stabilizers (Exterior) [Week 4]
Q: Evolution of parts and bodies [Week 4]
R: Cutting new legs, gears, and panels [Week 4]
S: Close-up of new gears and panel in action [Week 4]
T: Lady Strandbeest (before adding external dowel stabilizers) [Week 4]

Process Description

This immensely iterative process required a lot of forethought prior to laser cutting or even opening Illustrator. As mentioned above, we were inspired by the original Strandbeest and were lucky enough to find the measurements online. We took every number and multiplied it by 1/16 to reduce the size and make it more feasible for the purposes of the obstacle course. However, we quickly realized that there were a lot more challenges.

The first challenge we faced was the motorization and structure of this bot. We had to figure out quickly how to fit the motors into the main body as well as how to make the main body the minimum weight. We knew that the end product would still have fairly weak legs, so we wanted to make sure that the legs could support the body. Therefore, we had to redesign the box a little more during one of our iterations which would include cut-out patterns and that helped significantly reduce the weight (especially since the battery box was heavy).

The biggest challenge for the legs was the precision in measurements and designing appropriate gears. After our first designed gears failed, we used the online gear generator while also changing ratio to 2:1 to make the rotary movement's force stronger. We also had to make sure that there were no loose pieces in the box which required a few iterations as our designed required introducing new pieces continuously to increase stability.

The last challenge that we solved was adding more friction on the feet of the Lady Strandbeest and making the gear movement smoother by waxing. Lastly, we adjusted the angle of gears relative to each other and other sets of legs. This required modification in the code as we had to make sure that turning would move one leg forward and the other backward.

Reflection

Overall, this experience was really informative and rewarding: we were able to quickly solidify our vision and overcome challenges along the way that challenged our patience and loyalty to the original vision. Despite all the difficulties, every step involved a new challenge to solve which led us to the next iteration. We were able to continuously iterate to enhance the model and perfect the movement. As an example, we realized that we needed to add stabilizer for the dowels that connected all the legs across and we used zip ties to secure them in place and later we had to recreate the box with the slots for the laser cut piece to hold the dowel.

The biggest takeaway from this project was the power of persistence, the importance of planning and ensuring the equality of contribution of both members. We realized that we should have aimed to ensure modularity rather to avoid recreating the Lady Strandbeest every week. While this was tough for us after taking into account the limitation of time and resources, planning and distributing the work would have saved us a lot of time as we faced many unforeseen challenges along the way.

For future improvements, we'd like to ensure our Lady Strandbeest can surpass every obstacle in its way. This can be done by adding an extra set of legs on each side and using bigger screws and nuts for the bigger gears. We would also change the body to a smaller rectangular box by rearranging the placement of electronics. Lastly, we would like to experiment with other materials and combining them to examine stability, movement and aesthetic values. We discussed using materials such as acrylic and titanium but this would require some modifications as we must account for the behaviour and thickness of different materials.

Overall, we both learned a lot from taking this class and really appreciate all the help/support we've received during this journey!

Files

Code:

```
/*****************************************************************

 Mini Race Car kit -- Formula E Feather robot with HTU21D-F
 Temperature/Humidity sensor

 This is an example for our nRF51822 based Bluefruit LE modules

 Modified to drive a 3-wheeled BLE Robot Rover! by http://james.devi.to

 Pick one up today in the Adafruit shop!

 Adafruit invests time and resources providing this open source code,
 please support Adafruit and open-source hardware by purchasing
 products from Adafruit!

 MIT license, check LICENSE for more information
 All text above, and the splash screen below must be included in
 any redistribution
*****************************************************************/

#include <Arduino.h>
#include <Adafruit_BLE.h>
#include <Adafruit_BluefruitLE_SPI.h>

#include "BluefruitConfig.h"

#include <Adafruit_MotorShield.h>

#include <Wire.h>
// #include "Adafruit_HTU21DF.h"
// Connect Vin to 3V DC
// Connect GND to ground
// Connect SCL to I2C clock pin
// Connect SDA to I2C data pin

// Create the temp/humidity sensor object
// Adafruit_HTU21DF htu = Adafruit_HTU21DF();


// Create the motor shield object with the default I2C address
Adafruit_MotorShield AFMS = Adafruit_MotorShield();

// And connect 2 DC motors to port M3 & M4 !
Adafruit_DCMotor *L_MOTOR = AFMS.getMotor(3);
Adafruit_DCMotor *R_MOTOR = AFMS.getMotor(4);

//Name your RC here
String BROADCAST_NAME = "Strandbeest";

String BROADCAST_CMD = String("AT+GAPDEVNAME=" + BROADCAST_NAME);

Adafruit_BluefruitLE_SPI ble(BLUEFRUIT_SPI_CS, BLUEFRUIT_SPI_IRQ, BLUEFRUIT_SPI_RST);

// A small helper
void error(const __FlashStringHelper*err) {
  Serial.println(err);
  while (1);
```

```
}

// function prototypes over in packetparser.cpp
uint8_t readPacket(Adafruit_BLE *ble, uint16_t timeout);
float parsefloat(uint8_t *buffer);
void printHex(const uint8_t * data, const uint32_t numBytes);

// the packet buffer
extern uint8_t packetbuffer[];

char buf[60];

// Set your forward, reverse, and turning speeds
#define ForwardSpeed             255
#define ReverseSpeed             255
#define TurningSpeed             255


/**************************************************************************/
/*!
    @brief  Sets up the HW and the BLE module (this function is called
            automatically on startup)
*/
/**************************************************************************/
void setup(void) {
  Serial.begin(9600);
//  if (!htu.begin()) { //start the temp/humidity sensor
//    Serial.println("Couldn't find sensor!");
//    while (1);
//  }

  AFMS.begin();  // create with the default frequency 1.6KHz

  // turn on motors
  L_MOTOR->setSpeed(0);
  L_MOTOR->run(RELEASE);

  R_MOTOR->setSpeed(0);
  R_MOTOR->run(RELEASE);

  pinMode(buttonPin, INPUT);
  digitalWrite(buttonPin, HIGH);

  Serial.begin(115200);
  Serial.println(F("Adafruit Bluefruit Robot Controller Example"));
  Serial.println(F("-----------------------------------------"));

  /* Initialize the module */
  BLEsetup();
}

void loop(void)
{
  // read new packet data
  uint8_t len = readPacket(&ble, BLE_READPACKET_TIMEOUT);

  readController();
}


bool isMoving = false;
unsigned long lastPress = 0;

bool readController(){
  uint8_t maxspeed;
```

```
 // Buttons
  if (packetbuffer[1] == 'B') {

    uint8_t buttnum = packetbuffer[2] - '0';
    boolean pressed = packetbuffer[3] - '0';

    if (pressed) {
//      if(buttnum == 1){
//        ble.print("Temperature C: "); ble.println(htu.readTemperature());
//      }
//
//      if(buttnum == 2){
//        ble.print("Temperature F: "); ble.println((htu.readTemperature()*1.8)+32);
//      }
//
//      if(buttnum == 3){
//        ble.print("Humidity: "); ble.println(htu.readHumidity());
//      }

      if(buttnum == 4){

      }

      if(buttnum == 5){
        isMoving = true;
        L_MOTOR->run(BACKWARD);
        R_MOTOR->run(BACKWARD);
        maxspeed = ForwardSpeed;
        ble.println("Forward");
      }

      if(buttnum == 6){
        isMoving = true;
        L_MOTOR->run(FORWARD);
        R_MOTOR->run(FORWARD);
        maxspeed = ReverseSpeed;
        ble.println("Backward");
      }

      if(buttnum == 7){
        isMoving = true;
        L_MOTOR->run(BACKWARD);
        R_MOTOR->run(FORWARD);
        maxspeed = TurningSpeed;
        ble.println("Left");
      }

      if(buttnum == 8){
        isMoving = true;
        L_MOTOR->run(FORWARD);
        R_MOTOR->run(BACKWARD);
        maxspeed = TurningSpeed;
        ble.println("Right");
      }

      lastPress = millis();

      // speed up the motors
      for (int speed=0; speed < maxspeed; speed+=5) {
        L_MOTOR->setSpeed(speed);
        R_MOTOR->setSpeed(speed);
        delay(5); // 250ms total to speed up
      }
  } else {
```

```
    isMoving = false;
    // slow down the motors
    for (int speed = maxspeed; speed >= 0; speed-=5) {
      L_MOTOR->setSpeed(speed);
      R_MOTOR->setSpeed(speed);
      delay(5); // 50ms total to slow down
    }
    L_MOTOR->run(RELEASE);
    R_MOTOR->run(RELEASE);
   }
}
}

void BLEsetup(){
 Serial.print(F("Initialising the Bluefruit LE module: "));

 if ( !ble.begin(VERBOSE_MODE) )
 {
  error(F("Couldn't find Bluefruit, make sure it's in CoMmanD mode & check wiring?"));
 }
 Serial.println( F("OK!") );

 /* Perform a factory reset to make sure everything is in a known state */
 Serial.println(F("Performing a factory reset: "));
 if (! ble.factoryReset() ){
     error(F("Couldn't factory reset"));
 }

 //Convert the name change command to a char array
 BROADCAST_CMD.toCharArray(buf, 60);

 //Change the broadcast device name here!
 if(ble.sendCommandCheckOK(buf)){
  Serial.println("name changed");
 }
 delay(250);

 //reset to take effect
 if(ble.sendCommandCheckOK("ATZ")){
  Serial.println("resetting");
 }
 delay(250);

 //Confirm name change
 ble.sendCommandCheckOK("AT+GAPDEVNAME");

 /* Disable command echo from Bluefruit */
 ble.echo(false);

 Serial.println("Requesting Bluefruit info:");
 /* Print Bluefruit information */
 ble.info();

 Serial.println(F("Please use Adafruit Bluefruit LE app to connect in Controller mode"));
 Serial.println(F("Then activate/use the sensors, color picker, game controller, etc!"));
 Serial.println();

 ble.verbose(false);  // debug info is a little annoying after this point!

 /* Wait for connection */
 while (! ble.isConnected()) {
    delay(500);
 }

 Serial.println(F("*****************"));
```

```
  // Set Bluefruit to DATA mode
  Serial.println( F("Switching to DATA mode!") );
  ble.setMode(BLUEFRUIT_MODE_DATA);

  Serial.println(F("**********"));
}
```

Bibliography

We followed this tutorial for a mini race car for the first iteration of getting the code and instructions. We also used the reference slides given when soldering.

For the strandbeast, we used a lot of online resources to get inspiration and help:

- https://www.strandbeest.com/
- https://www.youtube.com/watch?v=D1tsRiUBesk
- https://www.youtube.com/watch?v=P0TQ5UiqaAs
- https://www.youtube.com/watch?v=JsWjM-IkNJM
- https://www.youtube.com/watch?v=C7FMIRfP1tk
- https://www.youtube.com/watch?v=C9UvdlX1bhs
- https://www.youtube.com/watch?v=m207F-6i8UI
- https://www.youtube.com/watch?v=WEwqiMcjLMU
- Simulation