DEADLINE OCTOBER 29, 2020

IEEEBSB DESIGN CONTEST

Design an automatic candy dispenser for trick-or-treaters that reduces the exposure and spread of COVID

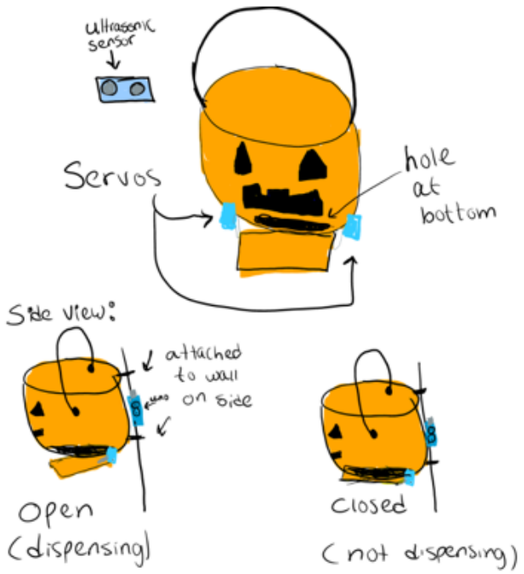
Rules:

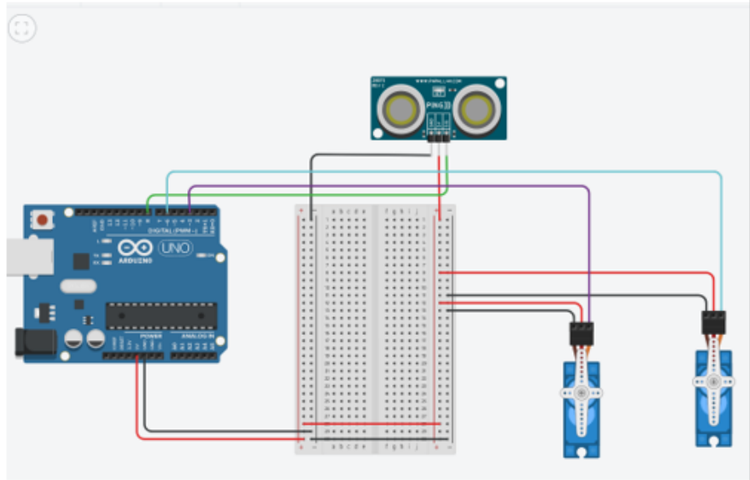
- Design must use a plastic pumpkin.
- Use of a microcontroller is allowed.
- Simulations are allowed.
- Building is encouraged.
- Must be a UTRGV student.

Win a halloween basket and points to redeem a prize of your choosing!

Submit your design showcase and any additional files that explains your design to IEEE.BTX@gmail.com

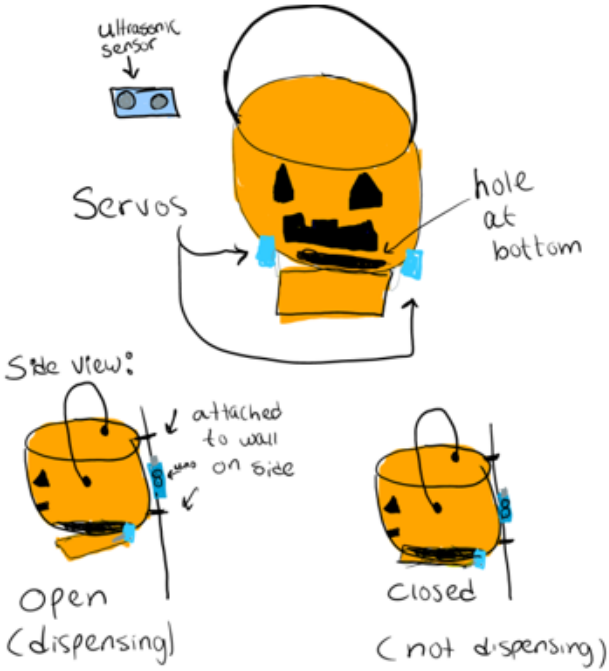
Example Submission:

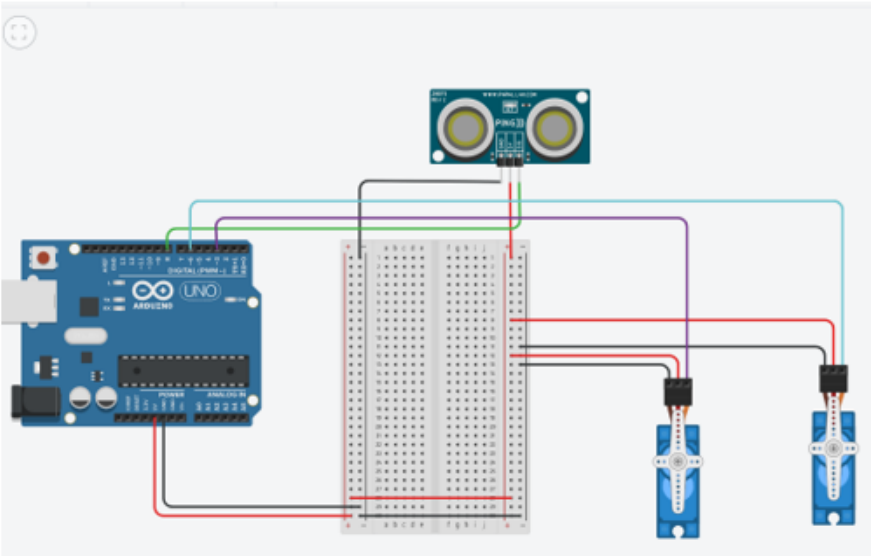




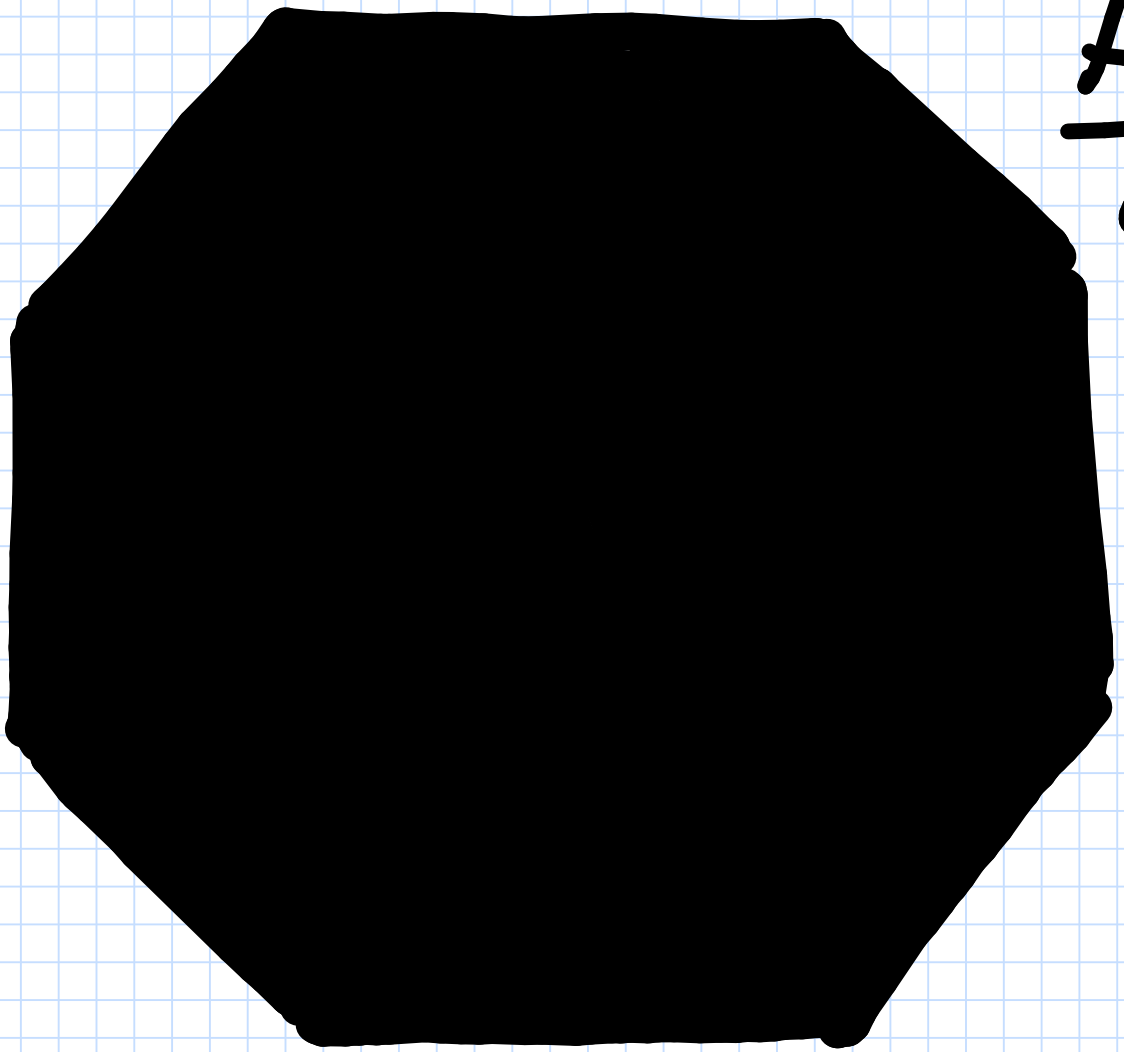
```
#include <Servo.h>

Servo servo1;
Servo servo2;

void setup()
{
  //Attach servos to pins 3&6
  servo1.attach(3);
  servo2.attach(6);
}

void loop()
{


  //Trigger sensor on
  pinMode(8,OUTPUT);
  digitalWrite(8, HIGH);
  delayMicroseconds(2);
  digitalWrite(8, LOW);

  //set pin as input & read pulse length from sensor
  pinMode(8, INPUT);
  long duration = pulseIn(8, HIGH);

  //Calculation of distance
  long distancecm = duration*0.034/2;

  //Detect person if closer less than 30cm
  if (distancecm<60)
  {
    /*Dispense candy and wait 30 seconds
    to give them time to close bag and leave
    and start detecting again*/
    servo1.write(90);
    servo2.write(90);
    delay(300);
    servo2.write(0);
    servo1.write(0);
    delay(30000);
  }


}
```

Example Submission:




UNO
ARDUINO
DIGITAL (PWM~)
POWER
ANALOG IN

```
#include <Servo.h>

Servo servo1;
Servo servo2;

void setup()
{
  //Attach servos to pins 3&6
  servo1.attach(3);
  servo2.attach(6);
}

void loop()
{


  //Trigger sensor on
  pinMode(8,OUTPUT);
  digitalWrite(8, HIGH);
  delayMicroseconds(2);
  digitalWrite(8, LOW);

  //set pin as input & read pulse length from sensor
  pinMode(8, INPUT);
  long duration = pulseIn(8, HIGH);

  //Calculation of distance
  long distancecm = duration*0.034/2;

  //Detect person if closer less than 30cm
  if (distancecm<60)
  {
    /*Dispense candy and wait 30 seconds
    to give them time to close bag and leave
    and start detecting again*/
    servo1.write(90);
    servo2.write(90);
    delay(300);
    servo2.write(0);
    servo1.write(0);
    delay(30000);
  }


}
```

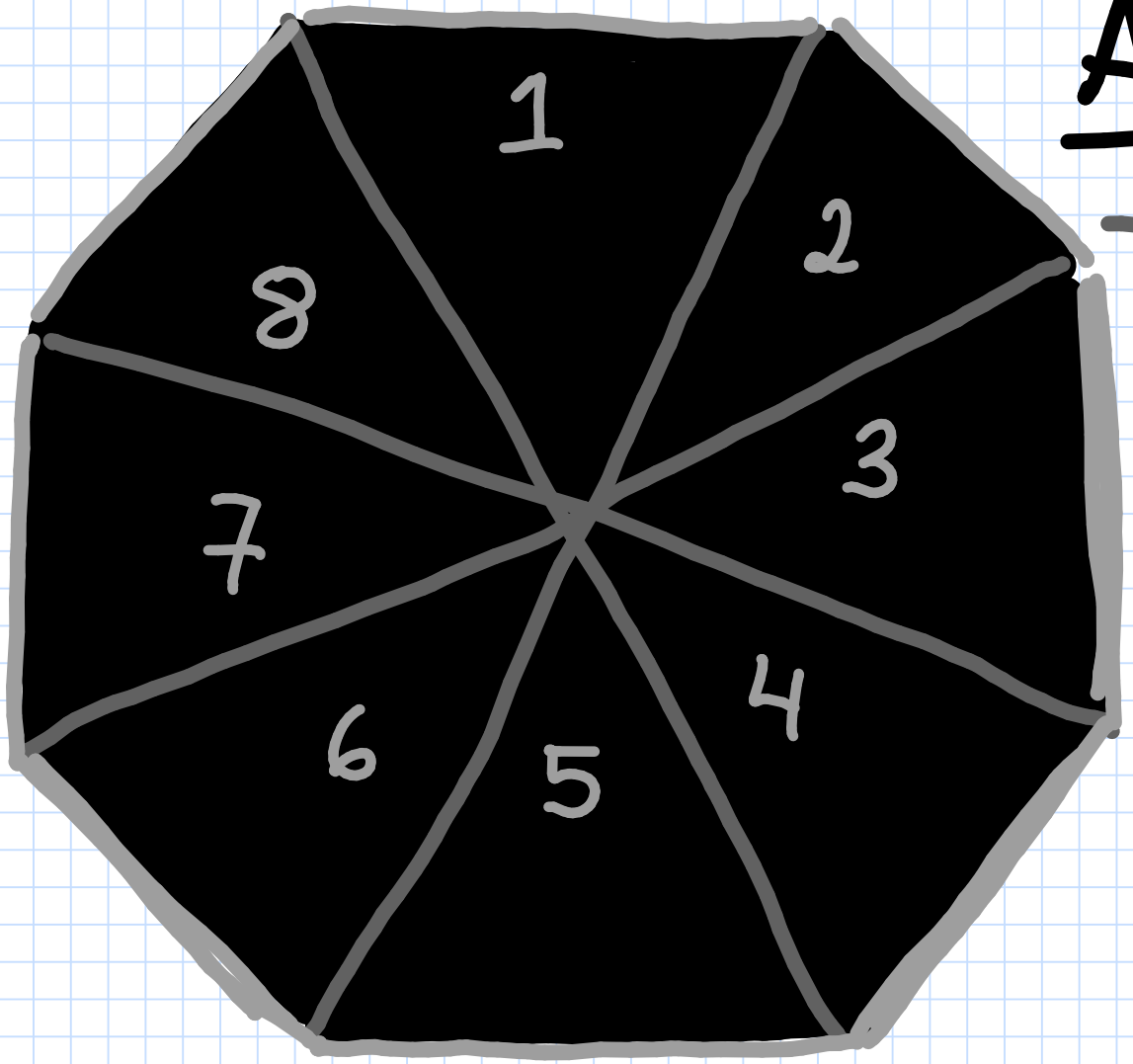


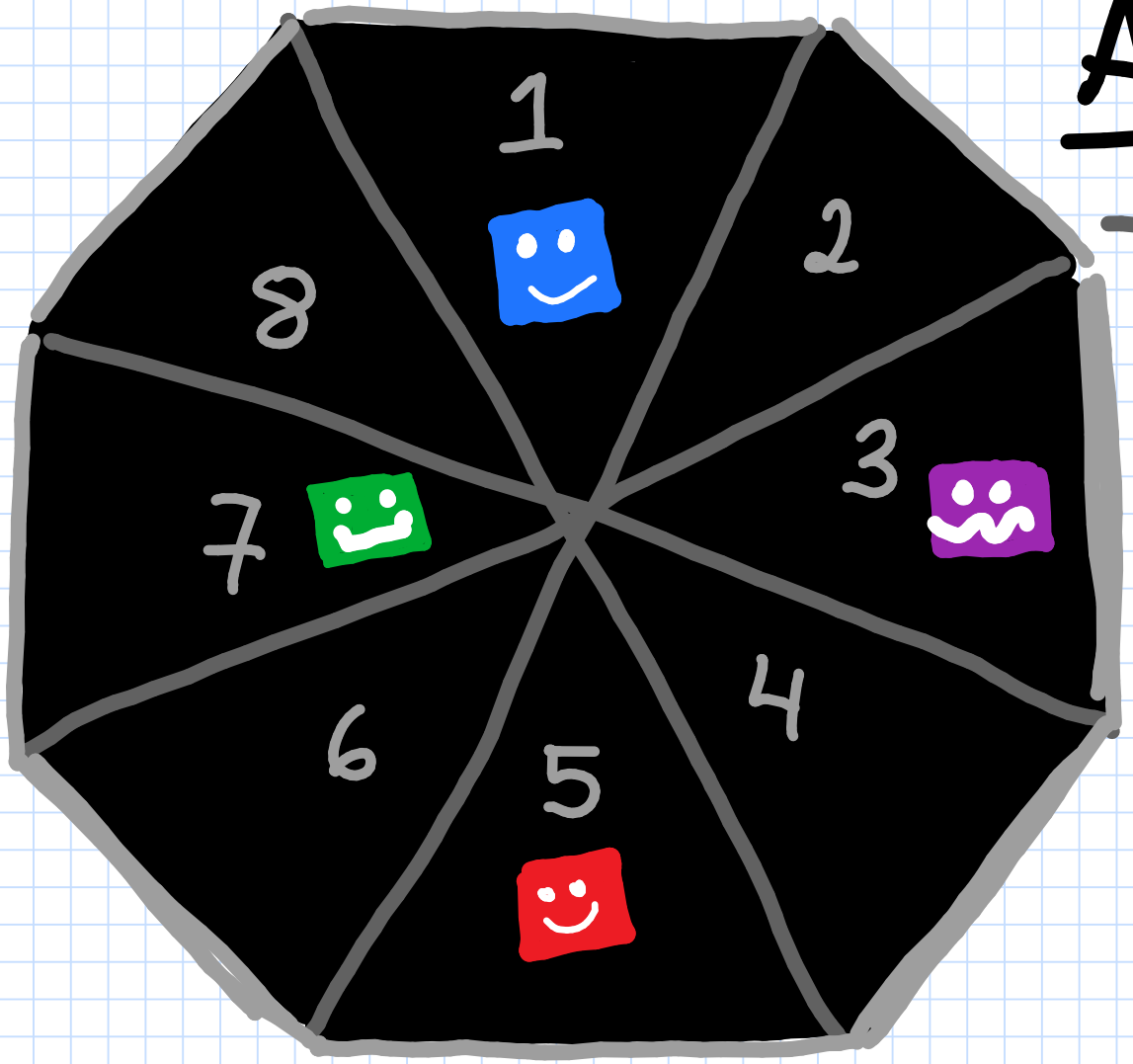
Arena

Chose octagon shape

Arena

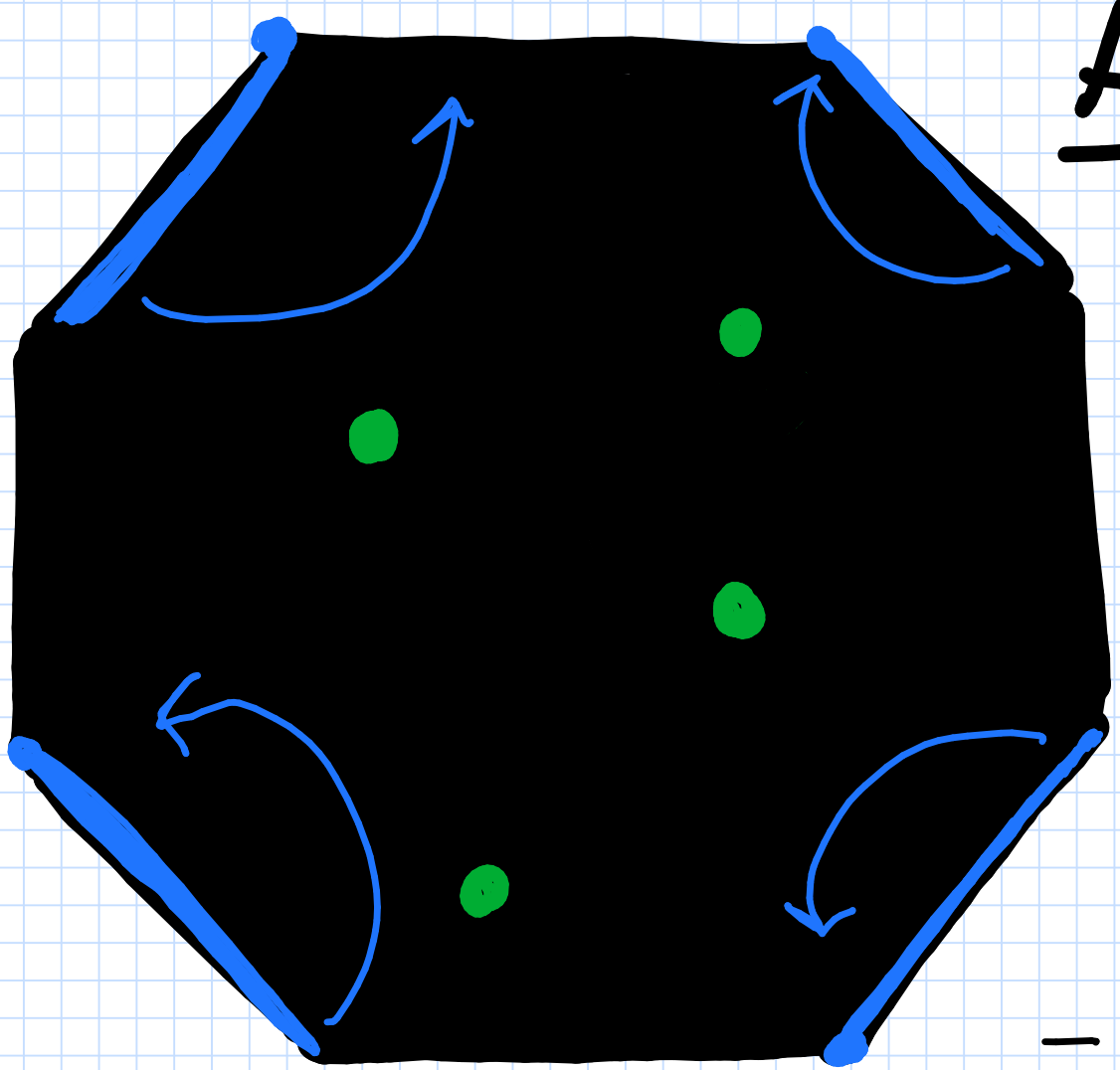
- segments





Arena

- segments
  - ↳ Placement



Arena

Hazards

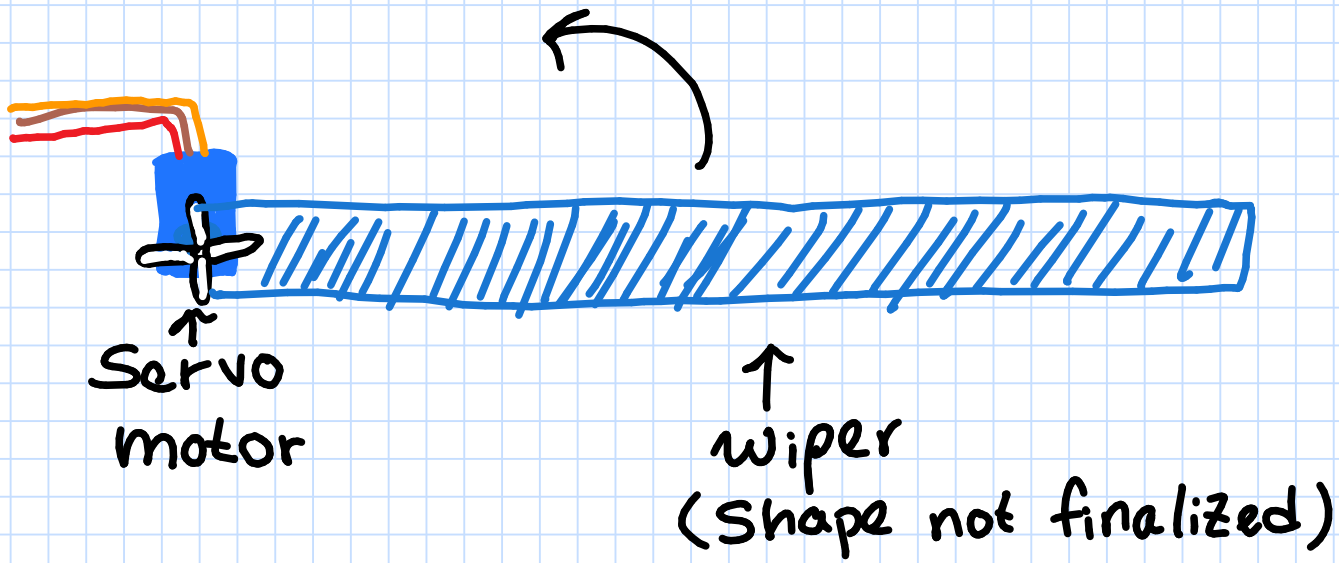
1. Wipers
   ↳ still to decide on length

2. rising pillars
   ↳ still to decide on size and length
   ↳ diameter

- How many?
- Placement

Servo ex:

Wiper mechanism



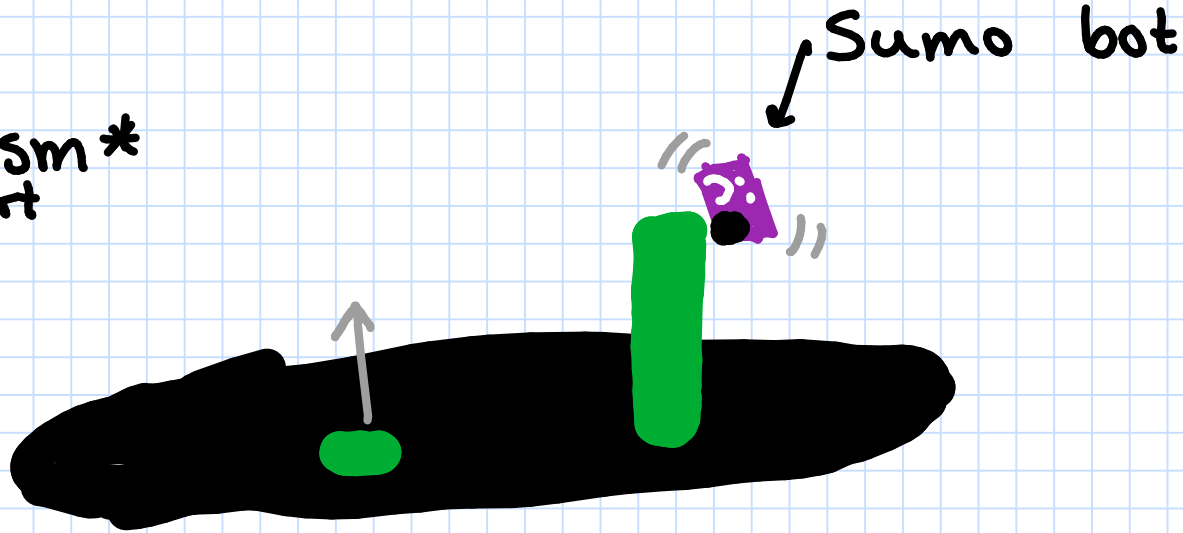
Things to consider:

- mechanical advantage
- servo torque rating
- sumo bot weight
- servo speed

Rising Pillars

To consider:

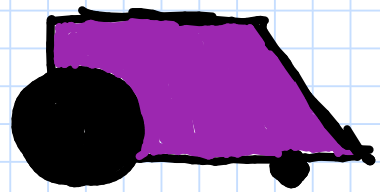
- total width
- rising distance
- rising mechanism*
  - ↳ weight support
- placement
- amount



Side view

Sumo bot

Side:



2x rear motorized wheels

1x pivot wheel for turning (no motor)

Features:

- LEDs (RGB) to switch based on color chosen by player

Donations

Severity of punishment is influenced by total amount donated, but not directly proportional

to do:

- Come up with equation
- make it general so we can tweak priority of punishments

to do while testing:

- rank severity of punishments / Hazards
  - ↳ pillars, wipers, reverse controls

ex:

If $10 are donated

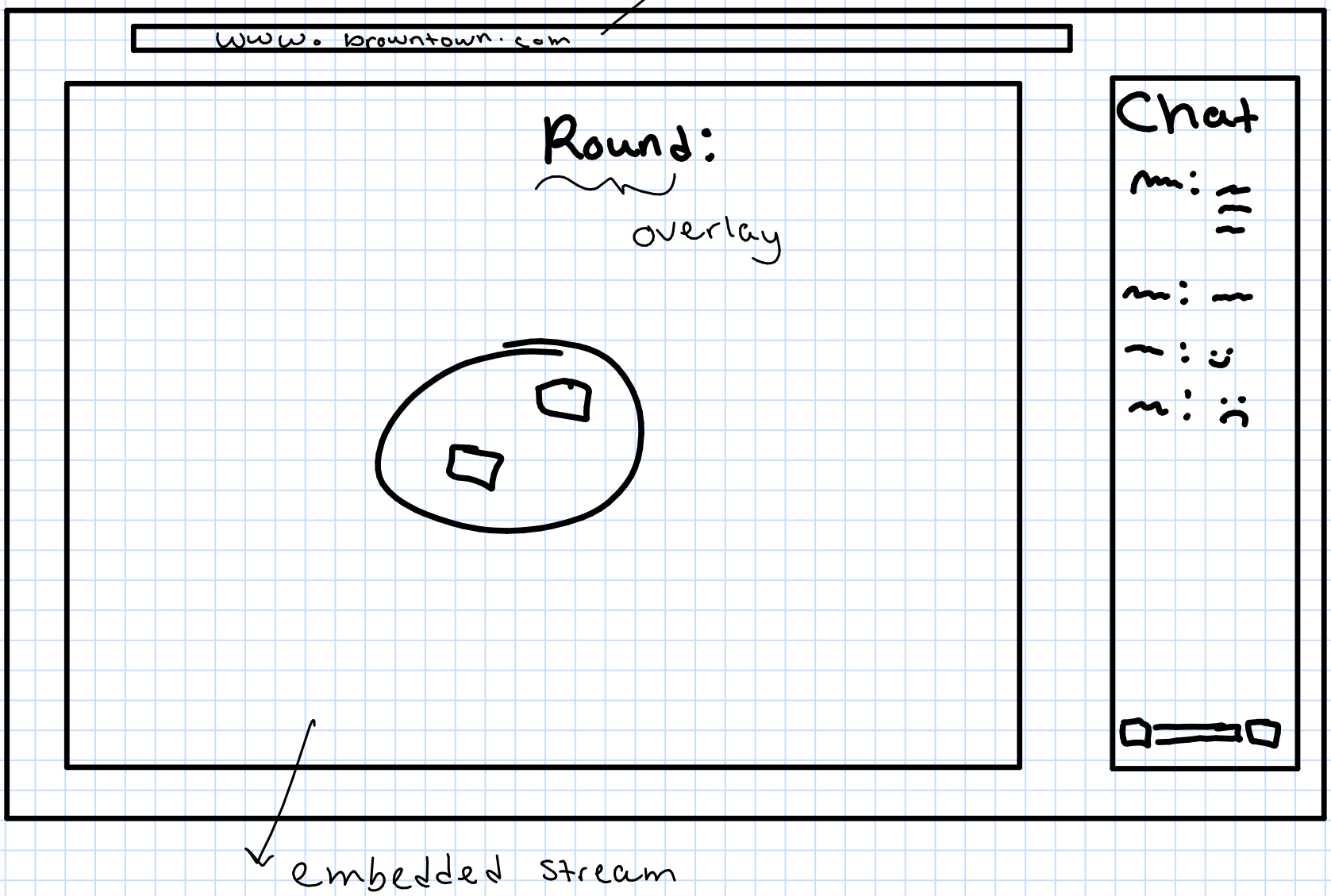
30% chance of pillars
20% chance of wipers
50% chance of reverse control

ex 2:

If $1 is donated

70% chance of pillars
25% chance of wipers
5% chance reverse controls

Web team



to do:

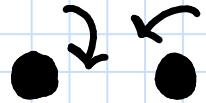
- Choose Streaming Platform
- Create basic control interface to esp-32
- set stream donation interactivity

General Programming

- main mcu: ESP-32
- Arduino IDE
  ↳ C/C++
- 3.3V Logic
- already supports wi-fi & bluetooth

to do:

- wheel movement (2x DC motors)

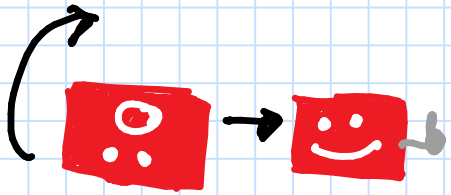


- receive request
  ↳ forward();
  backward();
  brake();

- design arena hazards
- get hardware list
- program arena hazards

Autonomous reset programming

- program autonomous routine after trigger
- Choose trigger type: hardware/software interrupts
- Choose sensors needed
- choose method of robot knowing its position
- ringout detection



Self-righting mechanism

Hardware QA

- gather list of hardware
  - ↳ order
- Choose battery type & voltage
  - ↳ 18650?
  - ↳ LiPO?
  - ↳ Li Ion?

  } Charging method
  - ↳ some require a separate charger
- Schematics
- datasheets
- assemble, solder, etc.

Art/UI

- Design logo: UTRGV <u>NOT</u> IEEE
  or a team name
  or BSB
- Color palette
- LED patterns
- warning messages/overlays

<u>ex:</u>

!!Activating Pillars!!