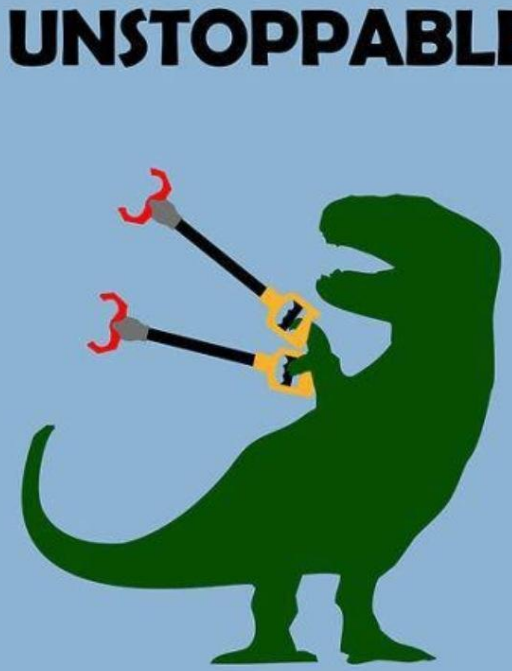


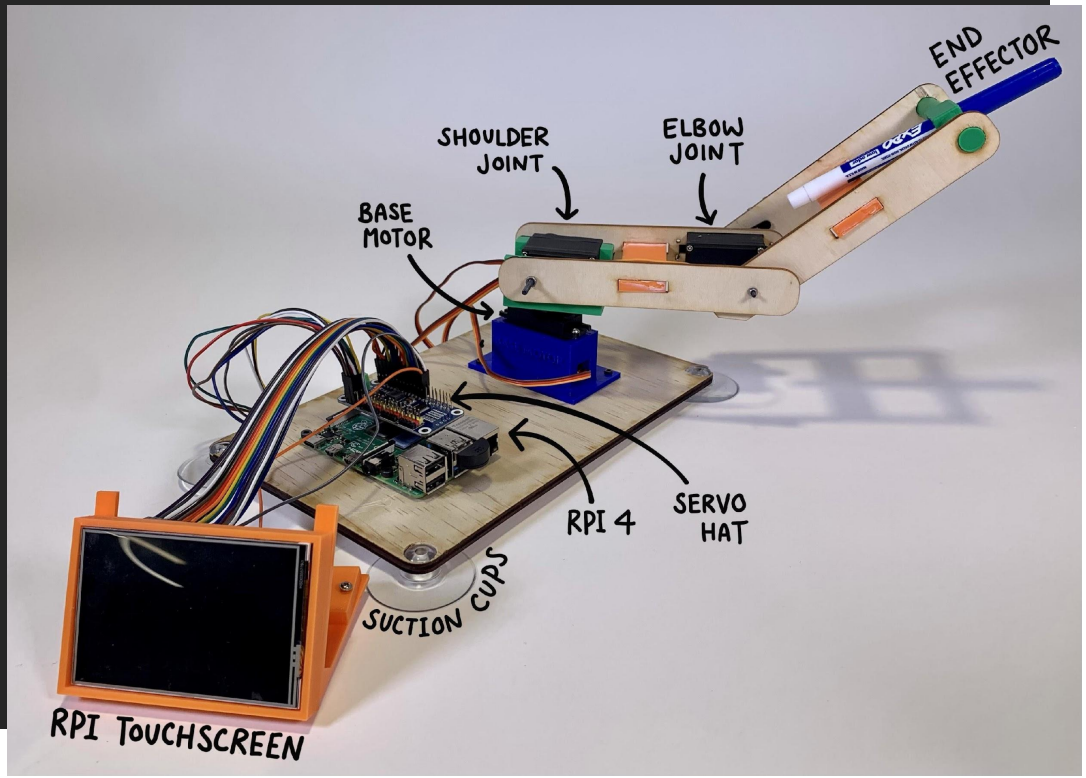
Assistive Technology for T-Rex's

Group #5

Sawyer Paccione, Matt Toven, Rebecca Shen

October 12th, 2021

Hardware

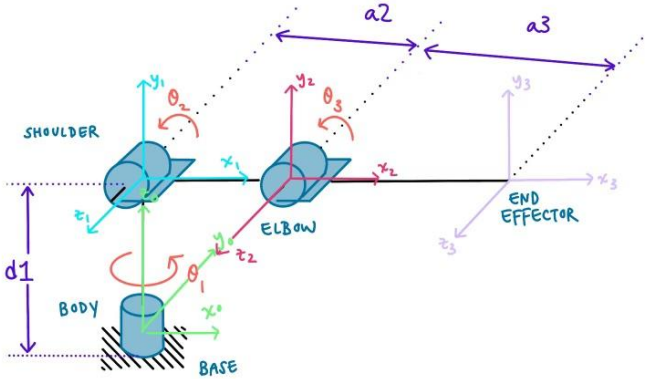


BoM

- ❑ Laser-cut wooden arms + baseplate
- ❑ 3D-printed orange arm supports & motor, touch screen, end effector holders
- ❑ Raspberry Pi 4
- ❑ 3 large servos
- ❑ RPi touchscreen hat module
- ❑ Jumper wires
- ❑ Suction cups

Inverse Kinematics

DH Transformation Matrices



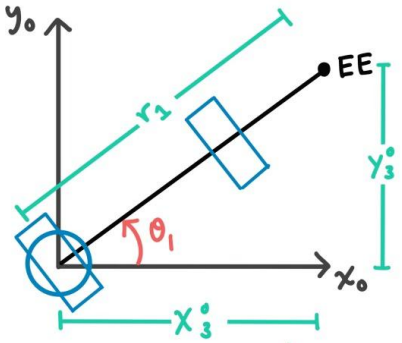
$$A_1^0 = \begin{bmatrix} c\theta_1 & 0 & s\theta_1 & 0 \\ s\theta_1 & 0 & -c\theta_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

	θ	α	a	d
1	θ_1	$\pi/2$	0	d_1
2	θ_2	0	a_2	0
3	θ_3	0	a_3	0

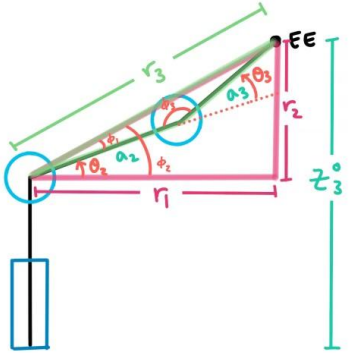
$$A_2^1 = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & a_2c\theta_2 \\ s\theta_2 & 0 & -c\theta_1 & a_2s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & a_3c\theta_3 \\ s\theta_3 & c\theta_3 & 0 & a_3s\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} x_3^0 \\ y_3^0 \\ z_3^0 \\ 1 \end{bmatrix} = \begin{bmatrix} c\theta_1(a_2c\theta_2 + a_3c(\theta_2+\theta_3) \\ s\theta_1(a_2c\theta_2 + a_3c(\theta_2+\theta_3) \\ a_3s(\theta_2+\theta_3) + a_2s\theta_2 + d_1 \\ 1 \end{bmatrix}$$

Graphical Approach



Top View



Side View

</> Software

5 Different Run Modes

1. Touch Screen
2. Initials Automation
3. Keyboard Control
4. Specific Angle Control
5. Single (x,y,z) Input

Robot Arm Class

- Control 3 MG996R Servo
- (x,y,z) → (θ1, θ2, θ3)
- User Interface
- Optimize Angle Choice
  - Validity
  - Closest

```
18
19  class RobotArm():
20      def __init__(self):
21          '''
22          Set channels to the
```

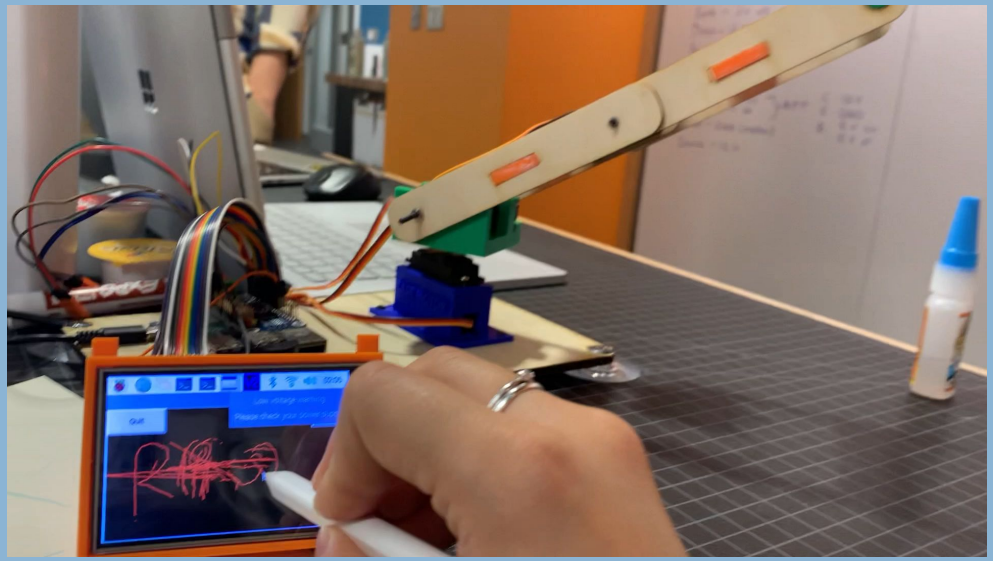
```
pi@raspberrypi:~/ME134/HW3 $ python3 main.py
Hello Potential User! How would you like to control the motor?
Type (1) for TouchScreen (2) for Keyboard or (3) for Initial Wr
itings (4) for UserInputTesting (5) for Coordinate Control?
```

Libraries Used

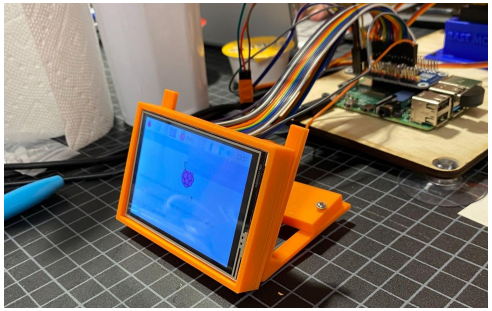
- Adafruit-circuitpython-servokit
- TKinter

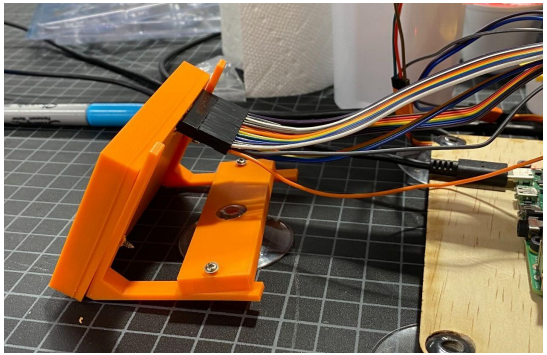
Creativity & Innovation

Touch Screen Display



- Touch screen display where you can draw the desired shape
- X, Y coordinates are tracked as you draw
- Coordinates are then plugged into inverse kinematics to find servo motor thetas





MATT

Wins

- Movement with 2 degrees of freedom
- We used the DH table to make transformation matrices and got theta equations
- We successfully were able to draw on touch screen & arm translates movement
- Keyboard Control

& Losses

- Not the movement we particularly wanted
- We could not successfully get correct angles to translate from our matrices/equations for theta 1,2,3 we calculated
- Doesn't draw very accurately
- Draws on the inside of a spherical shape now

Thank You!!