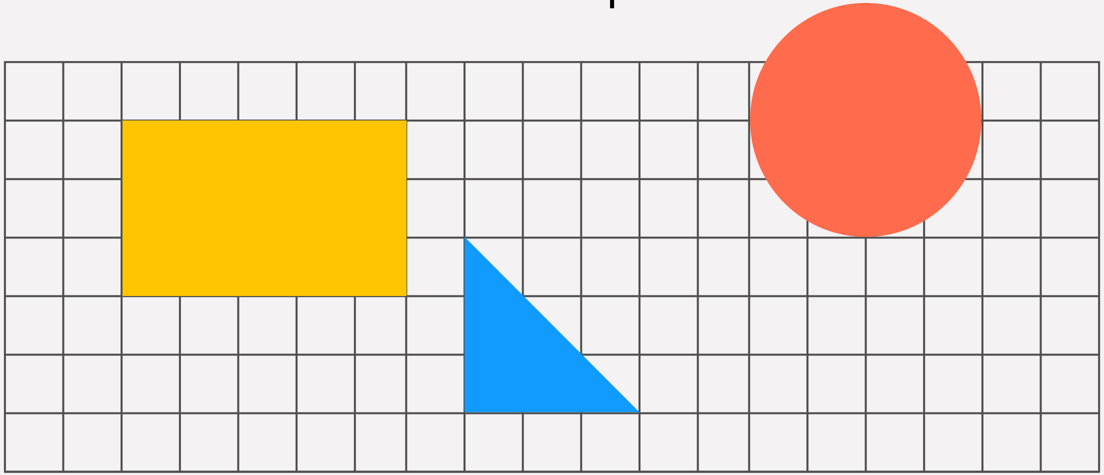
MAE 6291

Cocomelon Code

Aly Nguyen
GWU MAE Department



Introduction

Motivations

- Original Idea:
  - **Get a website deployed**
  - **Physical component: robotic arms hitting buttons**
  - **Digital component: clicker game**
- Continued Idea:
  - **Get a website deployed**
  - **Remotely activate Pi's sensors**
  - **Gashapon**

Goals

Personal

- Get GUI and motor practice

Project

- For impatient socially awkward Ipad babies in dermatology office waiting rooms that are unaware of the dangers of UV

Final

- Get project done and make better trapdoor

IoT Architecture

LAYER	THINGS
Interface Layer	• Linode (html and flask code deployed to)
Service Layer	• Flask (for backend of main website)
Network Layer	• Ngrok (similar to Cloudfare, for tunneling) • R.I.P. dweet.io
Sensing Layer	• Hardware - LED light • Hardware - Servo Motors (responding to cloud input)

Materials & Methods

Materials & Hardware Used

- 2 Servo Motors, 1 LED light, approx. 100 popsicle sticks, Glue gun & sticks, gashapon balls
- Raspberry Pi
- HyperText Transfer Protocol (HTTP), Internet Protocol (IP), Ngrok (Tunneling protocol)

App, or API developed

- Python & Libraries
  - **Flask (webserver), GPIO (pi pin control), Requests (connect devices), Time (timing of events),**

Methods:

- Flask website deployed through Linode, PuTTY (SSH terminal client), and WinSCP (secure file transfer)
- Remote RaspberryPi access through Ngrok
- LED light control & Servo motor control

Learning Outcomes

Conclusions:

- Websites are hard to deploy
- Not every python code can be pushed into a website
- Research library compatibility and updates (dweet.io)

Results:

- Downsized project due to time allotted
- More mechanically involved (Narins project review)
- Absolute disgust towards flask and in turn python

Future Directions

- Unused parts (Servo motors, LED lights)
- Graphics & Aesthetic HTML