



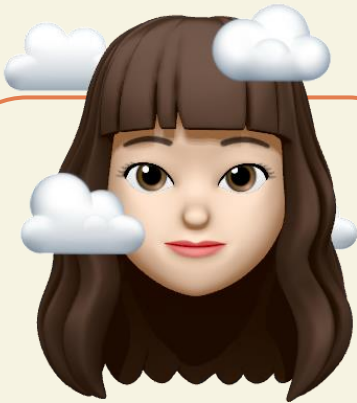
-
- 1 Leggo?
 - 2 팀원 소개
 - 3 플로우 차트
 - 4 진행 상황
 - 5 앞으로의 계획
 - 6 프로젝트 사이트
-



웹&앱을 이용한

거동 취약계층 맞춤형 길찾기 서비스와 탈부착 AI 보행 보조 디바이스

__Leggo__가 당신의 눈과 발이 되어드립니다.



유지원

팀장

CV & Raspberry-pi



노지민

팀원

Web Programming

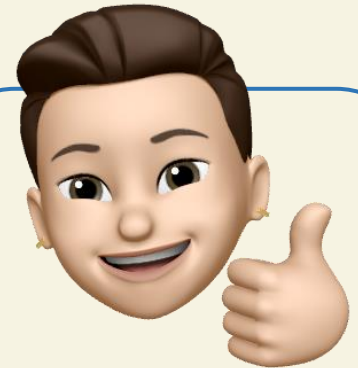


정진영

팀원

Arduino

App Programming



박무열

팀원

Arduino

App Programming





CV & Raspberry-pi

#1. YOLO v4-tiny test on Colab

- 경량화 모델인 YOLO v4-tiny 사용
- Darknet 프레임워크 사용
- Colab 환경에서 GPU 사용해 훈련
- 연습으로 사다리차에 대한 훈련 마치고 테스트

#1.



YOLO





CV & Raspberry-pi

#2. 도로 위 장애물 데이터셋 구성

- Selenium으로 연석 이미지 데이터 크롤링

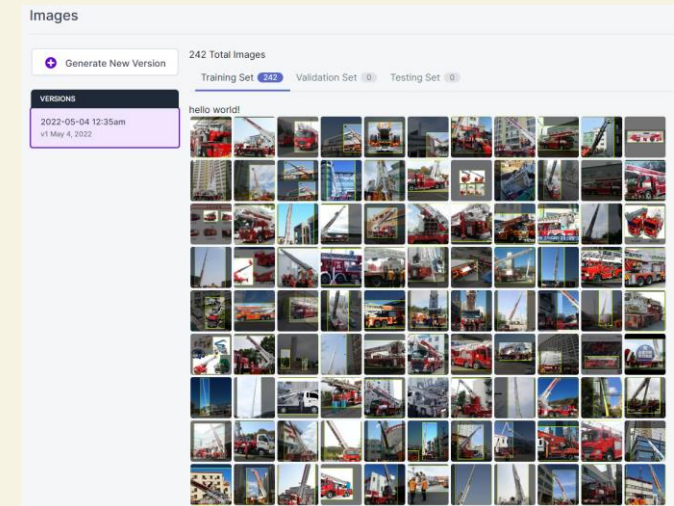
#3. 데이터셋 라벨링(Annotation)

- Roboflow 사이트 이용하여 데이터 라벨링

#2.

```
1 from selenium import webdriver
2 from selenium.webdriver.common.keys import Keys
3 import time
4 import urllib.request
5
6 driver = webdriver.Chrome()
7 driver.get("https://www.google.co.kr/img/hikoditahw&authuser=8log1")
8 elem = driver.find_element_by_name("q")
9 elem.send_keys("도로 건설")
10 elem.send_keys(Keys.RETURN)
11
12 SCROLL_PAUSE_TIME = 1
13 # Get scroll height
14 last_height = driver.execute_script("return document.body.scrollHeight")
15 while True:
16     # Scroll down to bottom
17     driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
18     # Wait to load page
19     time.sleep(SCROLL_PAUSE_TIME)
20     # Calculate new scroll height and compare with last scroll height
21     new_height = driver.execute_script("return document.body.scrollHeight")
22     if new_height == last_height:
23         try:
24             driver.find_element_by_css_selector("#my4qd").click()
25         except:
26             pass
```

#3.





CV & Raspberry-pi

#4. 아두이노 → 라즈베리파이 변경

- 아두이노에서는 CV 처리에 무리가 있다고 판단
- 라즈베리파이 4B로 변경

#5. 라즈베리파이 OS 설치 및 초기 부팅

- 카메라 명령 사용하기 위해 Bullseye → Buster 다운그레이드
- 라즈베리파이의 초기 설정 세팅 & SSH 연결
- 모니터 연결

#4.

	아두이노	라즈베리파이
중심 기능	외부기기 제어	데이터 처리
OS 설치 가능	X	Linux & ...
구분	마이크로 컨트롤러	마이크로 프로세서
GPU	X	O

#5.





Web Programming

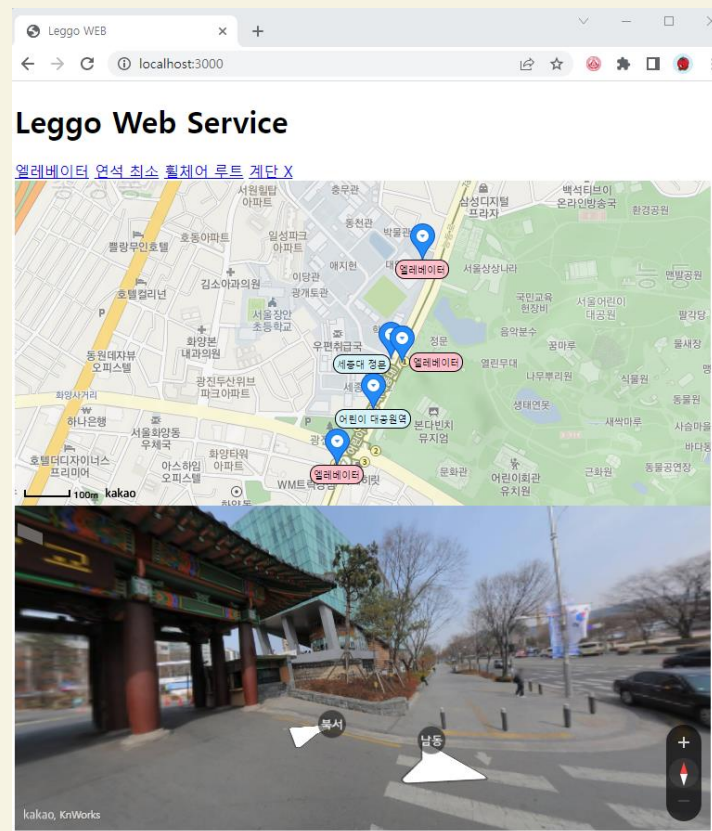
#1. 길찾기 서비스

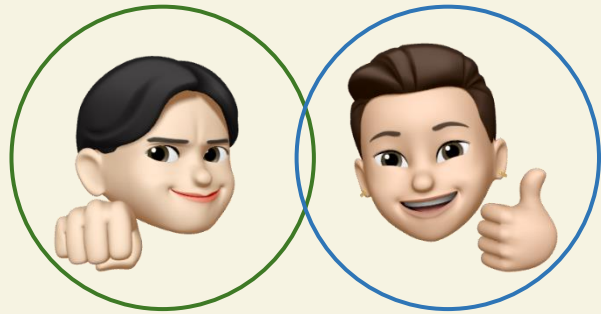
- 카카오 지도 API 이용
- 엘리베이터, 최소 연석, 휠체어 등 조건에 맞춘 서비스 제공

#2. 로드맵

- 도로의 시각적 정보 제공

#1.





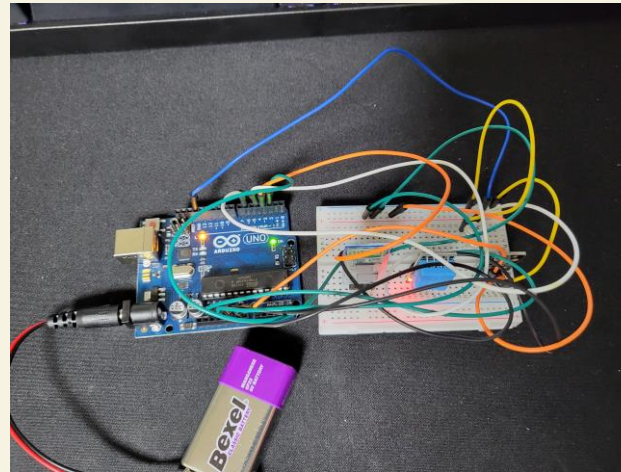
App Programming & Arduino

#1. 아두이노 회로

- 블루투스 모듈
- 온습도 센서
- 조도 센서
- LED

#2. 아두이노 스케치 코드

#1.



#2.

```
temphumidlight | 아두이노 1.8.19
파일 편집 스케치 툴 도움말
temphumidlight
SoftwareSerial HC06(2, 3); // RX, TX
int LIGHT_PIN = A0;
int LIGHT_THRESHOLD = 600;

void setup() {
  Serial.begin(9600); // pc - arduino
  HC06.begin(9600); //bluetooth - arduino
  dht.begin();
}

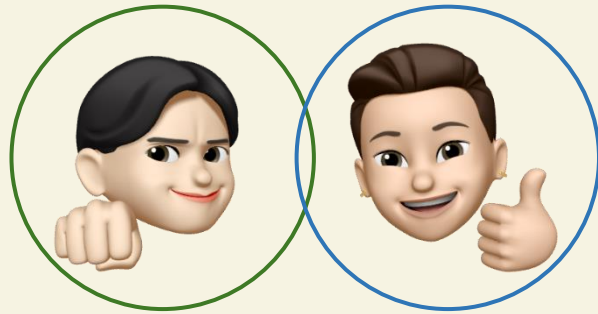
void loop() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  String data1 = String(h) + "," + String(t);
  HC06.print("T&H :");
  HC06.println(data1);

  Serial.print("HUMID&TEMP : ");
  Serial.print(h);
  Serial.print(",");
  Serial.println(t);

  delay(1000);
  // 조도 센서
  int val = analogRead(LIGHT_PIN);
  //VAL값이 크면 빛이 적음
  String data2 = String(val);
  HC06.print("L :");
  HC06.println(data2);

  Serial.print("LIGHT : ");
  Serial.println(val);
}
```



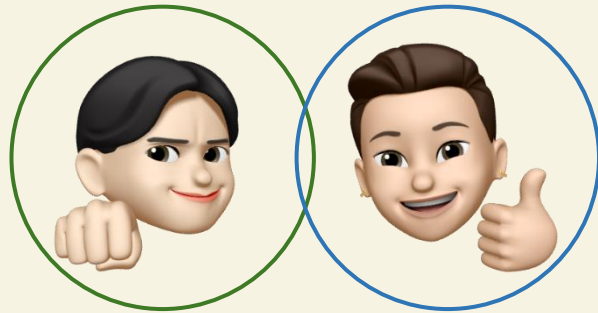
App Programming & Arduino

#3. 시리얼 모니터 출력

- 시간 별 온도 & 습도
- 시간 별 조도

#3.

```
22:22:23.317 -> HUMID&TEMP : 41.00,24.70
22:22:24.341 -> LIGHT : 1022
22:22:24.341 -> HUMID&TEMP : 41.00,24.70
22:22:25.373 -> LIGHT : 1022
22:22:25.419 -> HUMID&TEMP : 41.00,24.70
22:22:26.401 -> LIGHT : 1022
22:22:26.448 -> HUMID&TEMP : 41.00,24.70
22:22:27.428 -> LIGHT : 1022
22:22:27.474 -> HUMID&TEMP : 41.00,24.70
22:22:28.494 -> LIGHT : 1023
22:22:28.494 -> HUMID&TEMP : 41.00,24.70
22:22:29.524 -> LIGHT : 1022
22:22:29.571 -> HUMID&TEMP : 41.00,24.70
22:22:30.597 -> LIGHT : 1017
22:22:30.597 -> HUMID&TEMP : 41.00,24.70
22:22:31.624 -> LIGHT : 1022
22:22:31.670 -> HUMID&TEMP : 41.00,24.70
22:22:32.652 -> LIGHT : 1022
22:22:32.699 -> HUMID&TEMP : 41.00,24.70
22:22:33.680 -> LIGHT : 1022
22:22:33.727 -> HUMID&TEMP : 41.00,24.70
```

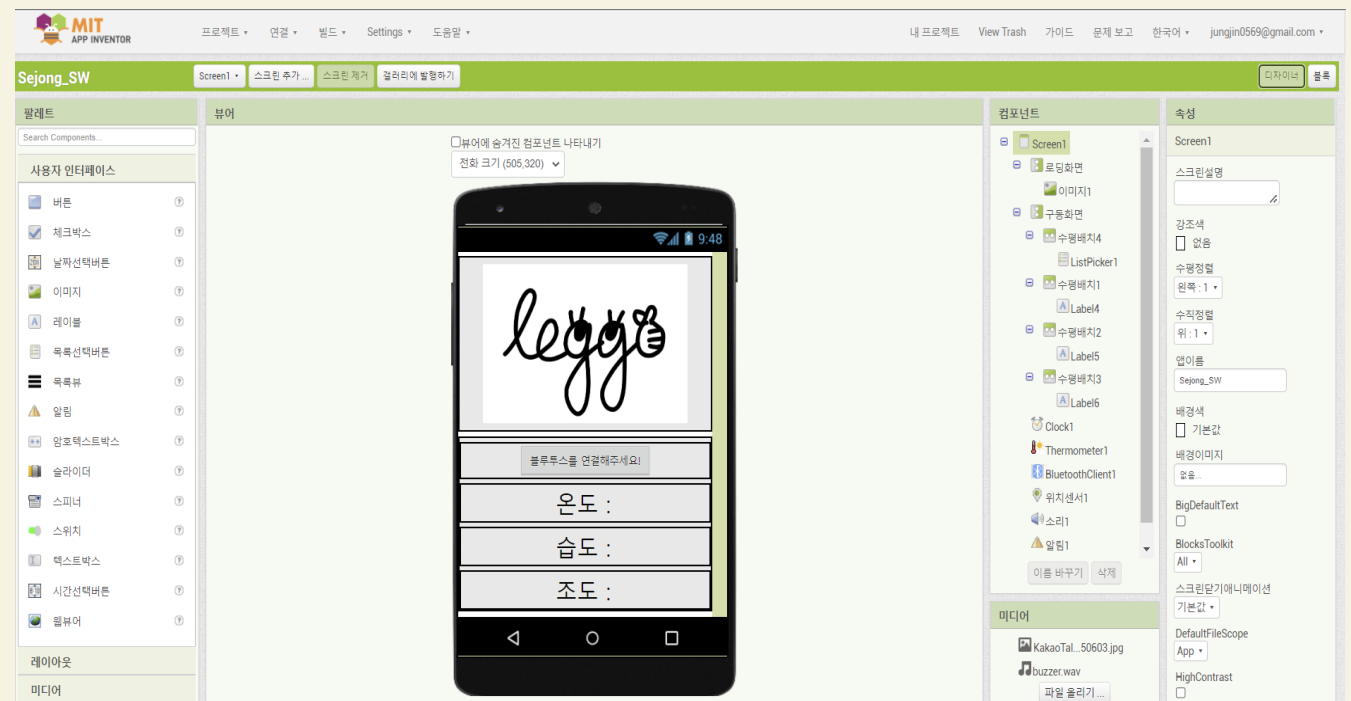


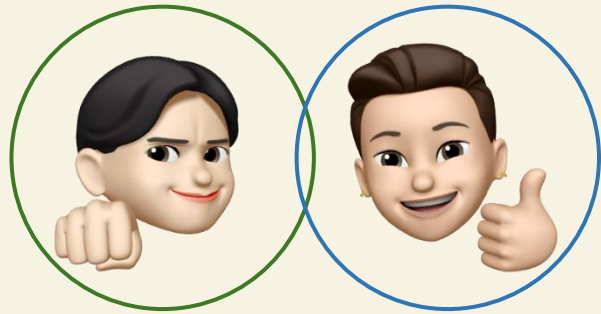
App Programming & Arduino

#4. 앱 프로그래밍 구현

- MIT App inventor2를 활용

#4.





App Programming & Arduino

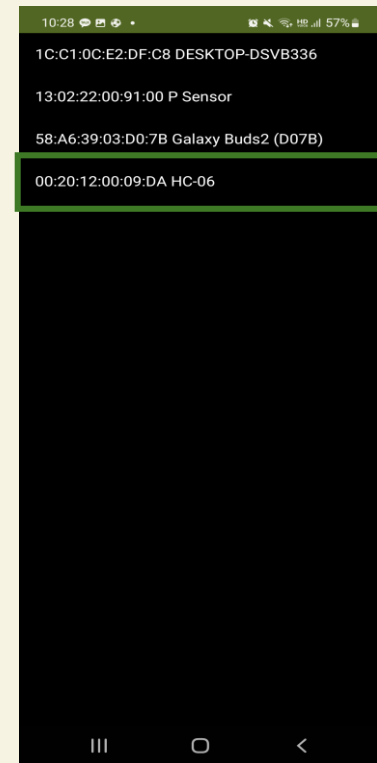
#5. 앱 실행

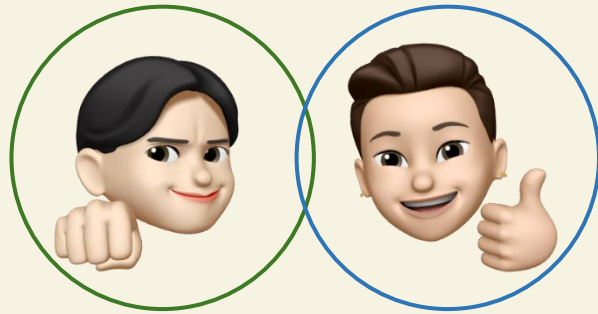
- 블루투스 연결

#5.



HC-06



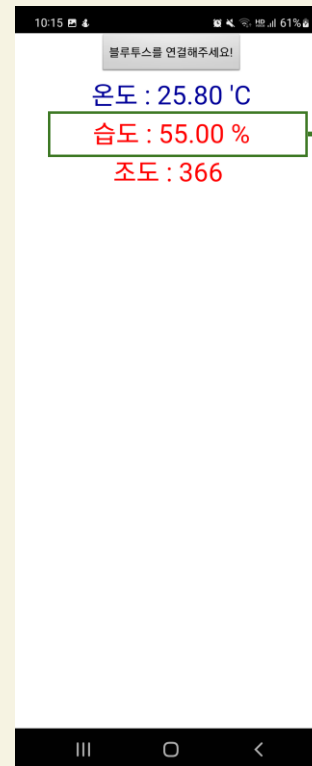


App Programming & Arduino

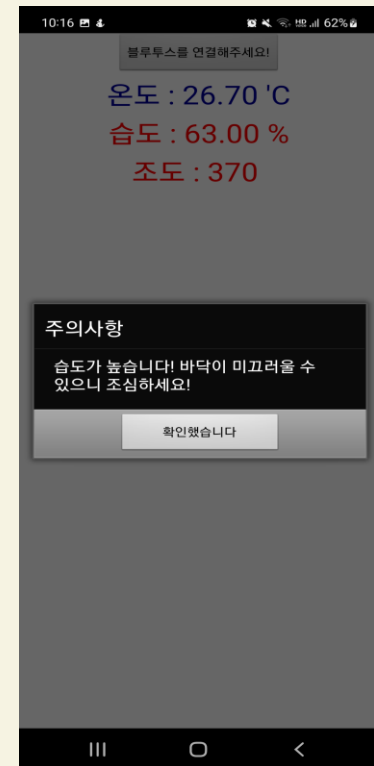
#6. 위험 시 반응

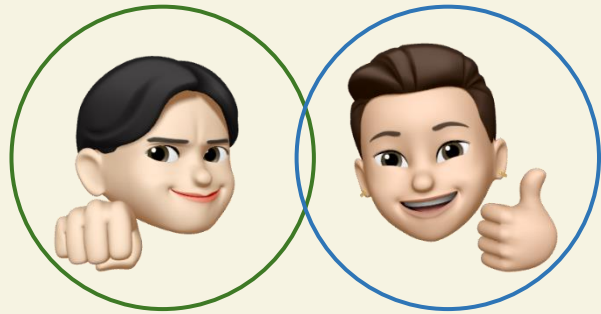
- 습도가 기준보다 높을 시, 경고 알람

#6.



위험한 환경 감지 - 빨간색



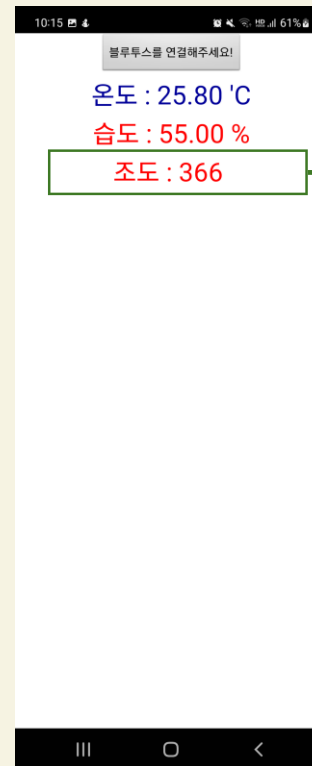


App Programming & Arduino

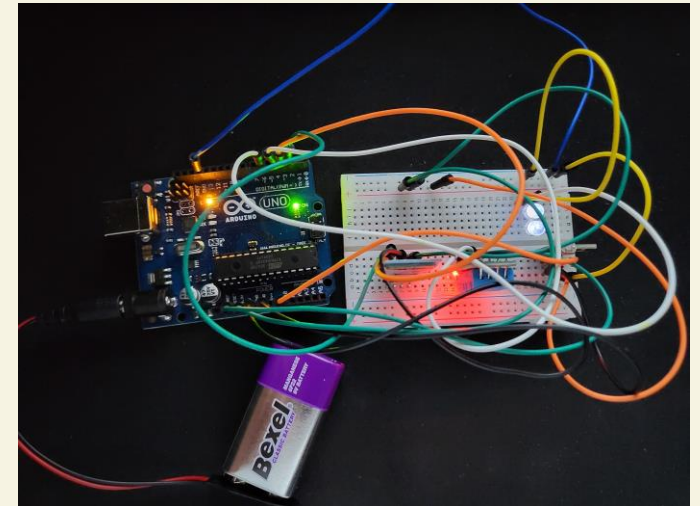
#7. 위험 시 반응

- 조도가 기준보다 낮을 시, LED 자동 점등
- 추후, 밝은 파워 LED 모듈로 대체할 예정

#7.



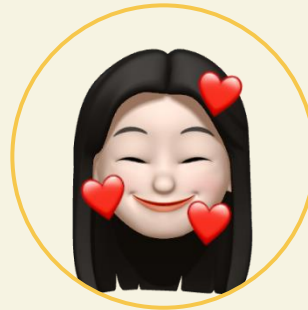
→ 위험한 환경 감지 - 빨간색





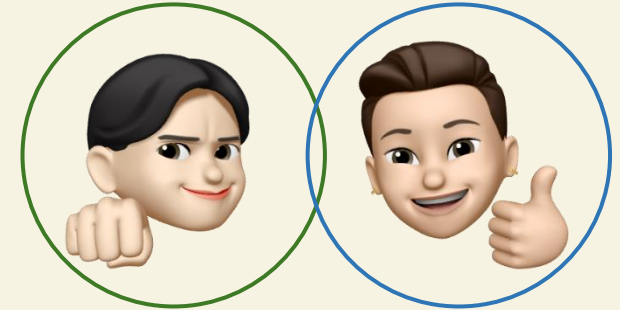
CV & Raspberry-pi

- 도로 위 장애물 데이터셋으로 YOLO V4-tiny custom
- 라즈베리파이에 YOLO 모델 데이터 임베드
- 카메라에서 가져온 영상(사진) 바로 YOLO 가능하도록
- Object Detection 결과 바탕으로 액추에이터 작동시키는 코드 작성



Web Programming

- css로 디자인 작업
- html, js 로직 구현 후 마무리
- 일주일 내에 완성 목표



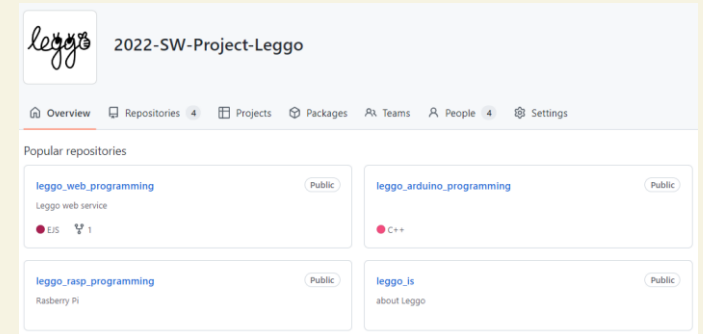
App Programming & Arduino

- 아두이노와 라즈베리파이의 크기가 구체화 되면 AutoCad를 이용해 3d프린터로 케이스 제작
- 고출력 파워 LED 받으면 일반 LED와 교체 후 출력되는 빛의 세기와 연결 확인



<https://github.com/2022-SW-Project-Leggo>

- Organization 조직하여 파트 별 레퍼지토리 관리
- 개발 코드 취합의 목적



<https://walkwithyou.notion.site/8f48671bdfa946d3ac30fc63b6083c80>

- 파트 별 작업 상황 공유, 일정 관리의 목적
- 기타 팀 자료 취합의 목적





감사합니다.