



토렌즈 포팅 매뉴얼

1 프로젝트 기술 스택

- A. FE
- B. BE
- C. IoT
- D. 기타

2 빌드 방법

- A. 백엔드 빌드 방법
- B. 프론트엔드 빌드 방법

3 AWS EC2 세팅

- 1. 서버 최신 상태 업데이트
- 2. Node.js 설치
- 3. Docker 설치
- 4. Docker 이미지 pull
- 5. Docker 이미지 실행

4 IoT 매뉴얼

- A. Jetson Nano OS 설치
- B. 필수 라이브러리 설치 (Mediapipe, OpenCV)
 - 1. 시스템 업데이트
 - 2. Mediapipe 및 TensorFlow 종속 패키지 설치
 - 3. opencv-python 설치
 - 4. RAM 스왑
 - 5. Mediapipe 설치
 - 6. Meiapipe-bin 업데이트
 - 7. 실행
- C. 카메라 연결 확인
 - 1. 카메라 포트 리스트 확인
 - 2. 카메라 작동 확인 및 얼굴 좌표 인식
- D. 블루투스 연결 확인

5 시나리오

- A. 대기 화면
- B. 시작 화면
- C. 안내 화면
- D. 진행 화면
- E. 보상 화면

1 프로젝트 기술 스택

A. FE

- 기술 스택(버전): React.js 18.2.0, react-dom 18.2.0, react-use-websocket 4.2.0, react-h5-audio-player 3.8.6, react-youtube 10.1.0
- 사용 툴: Visual Studio Code 1.74.2

B. BE

- 기술 스택(버전): Node.js 17.0-LTS, Express.js 4.18, ws 8.12
- 사용 툴: Visual Studio Code 1.74.2, Mobaxterm

C. IoT

- 기술 스택(버전): Python3 3.6.9, NVIDIA Jetson Nano Ubuntu 18.04, OpenCV 4.7.0, mediapipe 0.8.5-cuda102, websockets 1.3.1, PyAutoGUI 0.9.53
- 사용 툴: Visual Studio Code 1.74.2, Mobaxterm

D. 기타

- 형상 관리: GitLab
- 이슈 관리: Jira
- 커뮤니케이션: Notion, Mattermost
- 디자인: Figma, Adobe Photoshop 2022

2 빌드 방법

A. 백엔드 빌드 방법

- Commadn Shell을 통해 프로젝트 폴더 안의 ServerNode 폴더 안으로 이동
- dockerfile 제작

```
From node:17
WORKDIR /use/app
COPY package*.json ./

RUN npm install

COPY . ./
EXPOSE 8000
CMD ["node", "index.js"]
```

- docker 이미지 빌드

```
docker bulid -t choiseonho/node_server
```

- docker push

```
docker push choiseonho/node_server
```

B. 프론트엔드 빌드 방법

- Commadn Shell을 통해 프로젝트 폴더 안의 frontend\ 폴더 안으로 이동
- dockerfile 제작

```
From node:17
WORKDIR /use/app
COPY package*.json ./
```

```
RUN npm install

COPY . ./
EXPOSE 3000
CMD ["npm", "start"]
```

3. docker 이미지 빌드

```
docker build -t choiseonho/front
```

4. docker push

```
docker push choiseonho/front
```

3 AWS EC2 세팅

1. 서버 최신 상태 업데이트

```
sudo apt-get update
```

2. Node.js 설치

```
sudo apt-get install nodejs
```

3. Docker 설치

```
sudo apt-get install docker
```

4. Docker 이미지 pull

```
docker pull choiseonho/front
```

```
docker pull choiseonho/node_server
```

5. Docker 이미지 실행

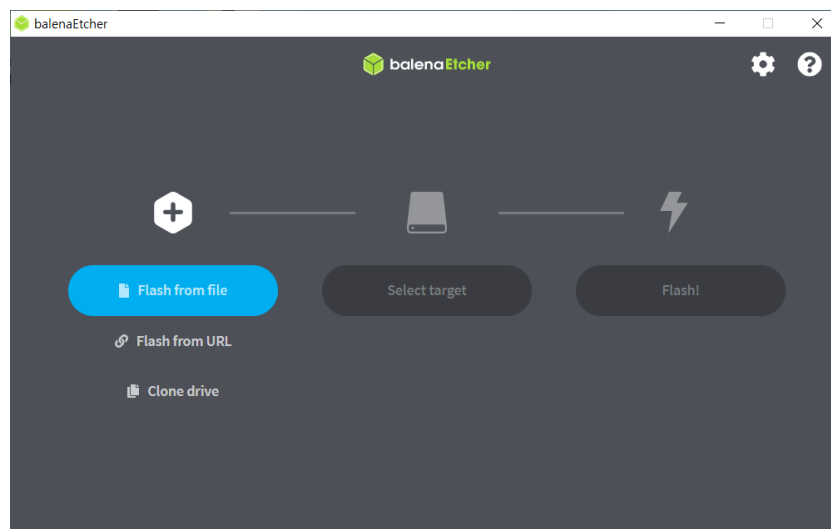
```
docker run -p 80:3000 -t choiseonho/front
```

```
docker run -p 8000:8000 -p 8002:8002 -p 8003:8003 -t choiseonho/node_server
```

4 IoT 매뉴얼

A. Jetson Nano OS 설치

balenEtcher 프로그램을 사용하여 OS 설치



B. 필수 라이브러리 설치 (Mediapipe, OpenCV)



기본 설치 방식과 달리 Jetson Nano에서는 다른 방식으로 설치

1. 시스템 업데이트

```
sudo apt update
```

2. Mediapipe 및 TensorFlow 종속 패키지 설치

1. TensorFlow에 필요한 시스템 패키지 설치

```
sudo apt-get update
sudo apt-get install libhdf5-serial-dev hdf5-tools libhdf5-dev zlib1g-dev zip libjpeg8-dev liblapack-dev libblas-dev gfortran
```

2. pip3 설치 및 업그레이드

```
sudo apt-get install python3-pip
sudo pip3 install -U pip testresources setuptools==49.6.0
```

3. Python 종속 패키지 설치

```
sudo pip3 install -U --no-deps numpy==1.19.4 future==0.18.2 mock==3.0.5 keras_preprocessing==1.1.2 keras_applications==1.0.8 gast=
sudo env H5PY_SETUP_REQUIRES=0 pip3 install -U h5py==3.1.0
```

3. opencv-python 설치

```
sudo apt-get install python3-opencv
sudo apt-get remove python3-opencv
```

4. RAM 스왑

```
git clone https://github.com/JetsonHacksNano/installSwapfile.git
cd installSwapfile
./installSwapfile.sh
```

5. Mediapipe 설치

```
cd ~
git clone https://github.com/google/mediapipe.git
cd mediapipe
sudo apt-get install -y libopencv-core-dev libopencv-highgui-dev libopencv-calib3d-dev libopencv-features2d-dev libopencv-imgproc-dev
sudo chmod 744 setup_opencv.sh
./setup_opencv.sh
```

6. Mediapipe-bin 업데이트

```
sudo pip3 install opencv_contrib_python
sudo apt install curl
cd Downloads
sudo apt install unzip
unzip mediapipe-bin.zip
```

7. 실행

```
cd mediapipe-bin
sudo pip3 install numpy-1.19.4-cp36-none-manylinux2014_aarch64.whl mediapipe-0.8.5-cuda102-cp36-none-linux_aarch64.whl
pip3 install dataclasses
```

C. 카메라 연결 확인

1. 카메라 포트 리스트 확인

```
v4l2-ctl --list-devices
```

```
C922 Pro Stream Webcam (usb-70090000.xusb-2.4):  
/dev/video0
```

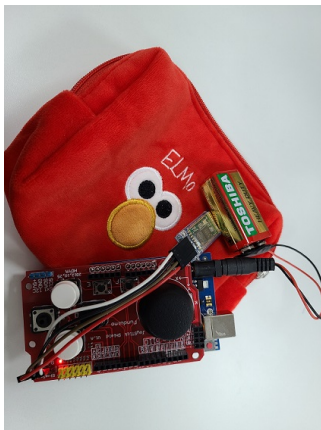
2. 카메라 작동 확인 및 얼굴 좌표 인식

```
python3 face-detect.py
```



D. 블루투스 연결 확인

```
python3 bluetooth_btn.py
```



```
c203@c203-desktop: ~  
c203@c203-desktop:~$ python3 bluetooth_btn.py  
bluetooth connected!  
receive : OK
```

5 시나리오

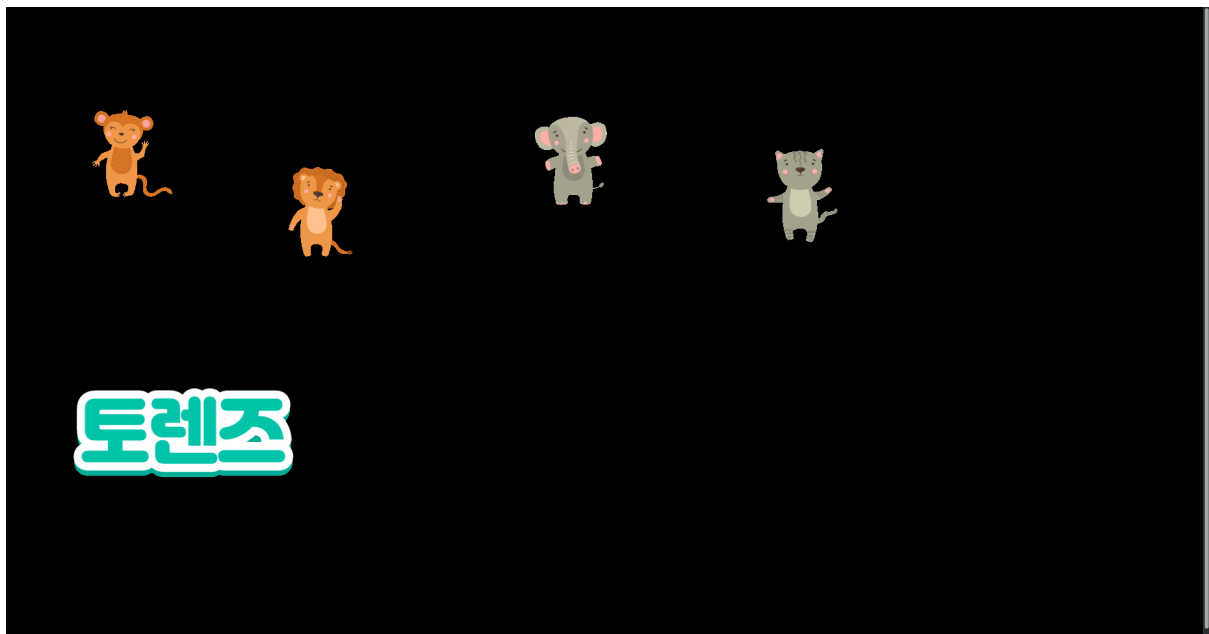
A. 대기 화면

2시 37분 39초



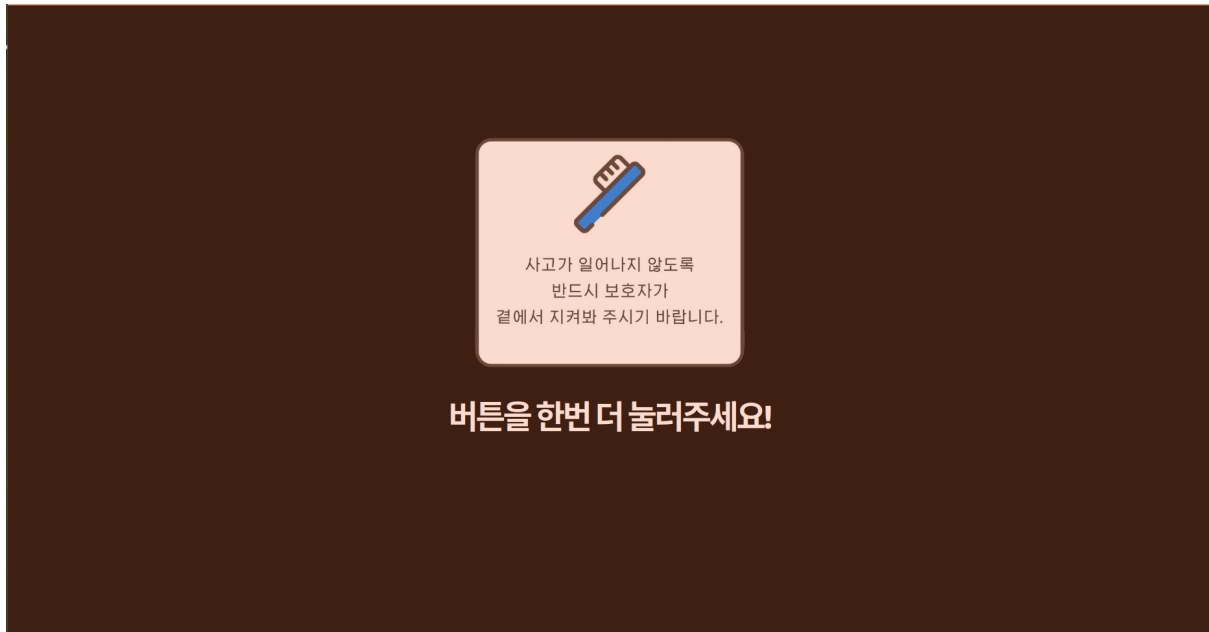
- 지정된 양치 시간에 얼굴 인식이 되면 **B. 시작화면**으로 전환된다.

B. 시작 화면



- 블루투스 버튼 입력 대기 (1분 30초)
 - 1분 30초 내 버튼을 누르지 않으면, **A. 대기화면**으로 전환된다.
 - 버튼 입력이 확인되면, 다음 화면 **C. 안내 화면**으로 전환된다.

C. 안내 화면



- 사고 방지를 위해 보호자가 곁에서 지켜봐 달라는 안내 문구 확인 가능하다.
- 안내 문구 확인 후, 블루투스 버튼 입력을 하게 되면 **D. 진행 화면**으로 전환된다.

D. 진행 화면



1. 세균



2. 반짝

- **좌측**에는 양치 방법에 대한 교육 영상이 재생된다.
- **우측**에는 아동의 얼굴을 인식, AR 기술을 활용하여, 아동의 흥미 유발을 유도한다.
- 아동의 양치 진행도에 따라 세균이 사라지는 효과를 볼 수 있다. (**1. 세균** → **2. 반짝**)
- 양치를 완료하면, **E. 보상 화면**으로 전환된다.

E. 보상 화면



- 아동의 보상 심리를 자극하기 위해 참! 잘했어요 도장 출력
- 1회 도장 출력 후 **A. 대기 화면**으로 전환된다.