



0419 진행현황



TEAM B

1

Robot Call

1 Robot Call

● Mode

Idle Mode

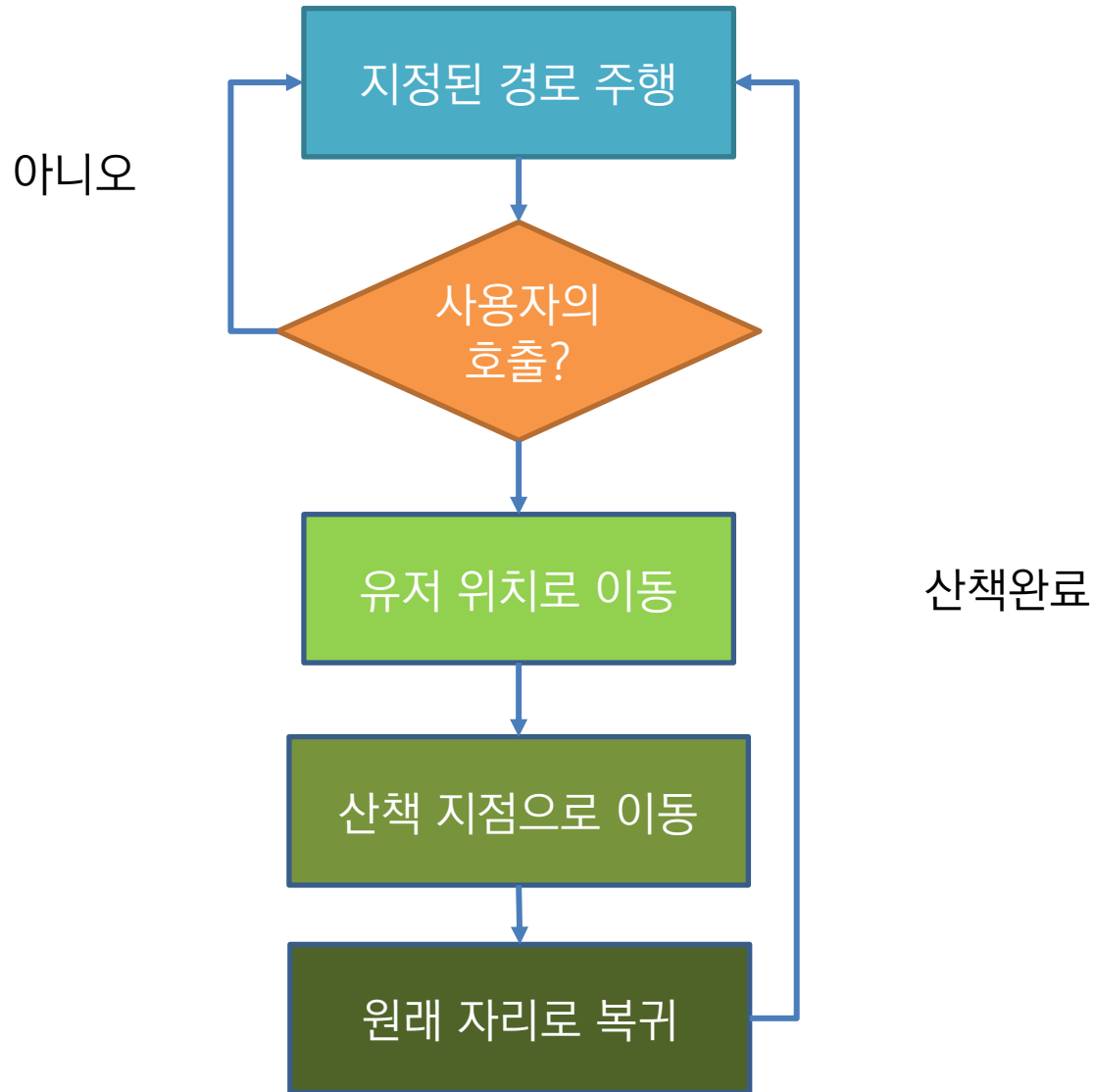
평상시에는 병원에서 지정한
경로대로 로봇이 움직이도록
설정

Active Mode

유저가 로봇을 부를 위치와,
산책할 경로를 지정해주면 로
봇이 사용자의 위치까지 온 후
같이 산책을 하도록 설정.

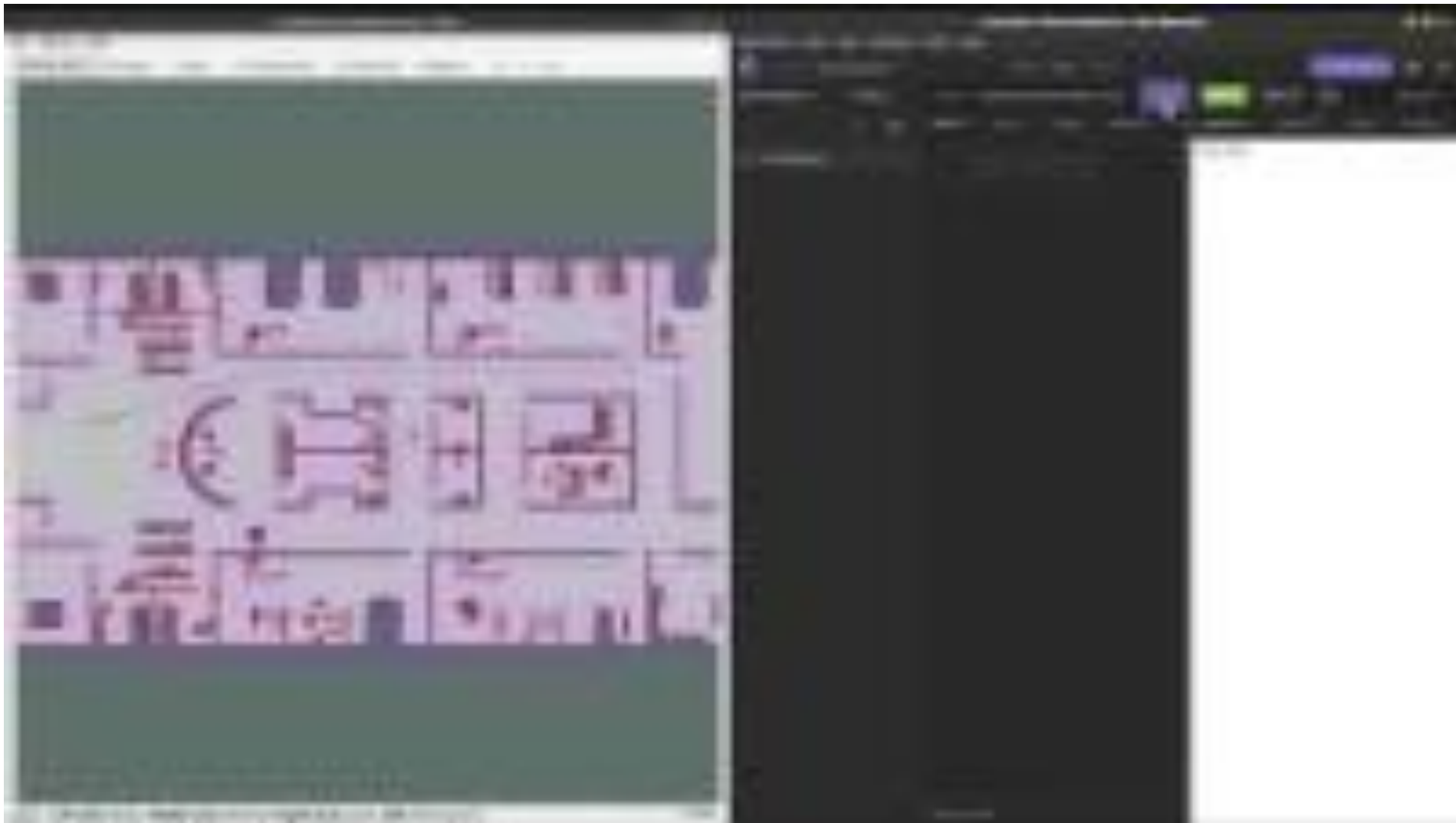
1 Robot Call

● Flow



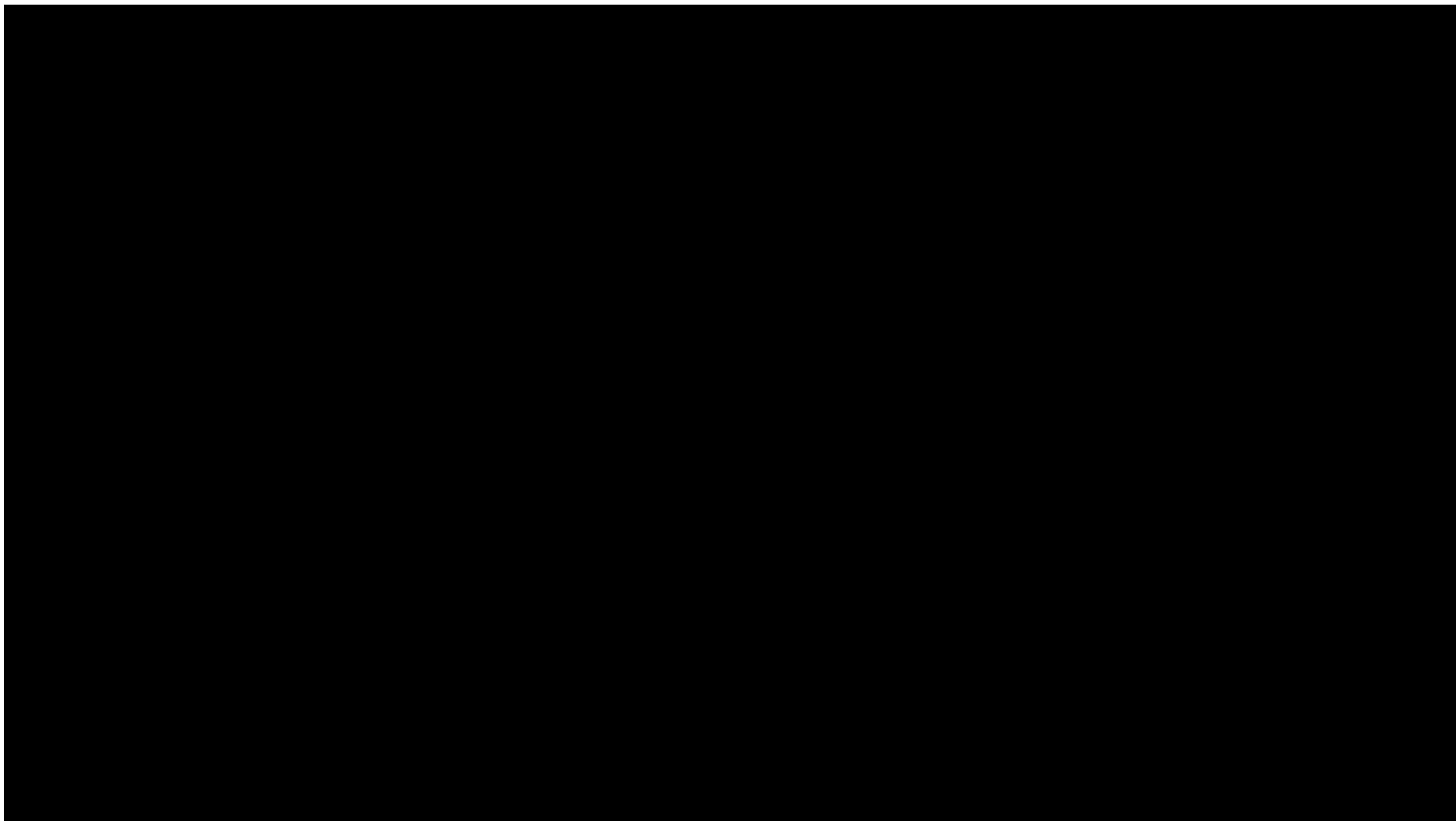
1 Robot Call

- Example



1 Robot Call

- Example(2)



1 Robot Call

● Difficulty

현재의 문제점

API를 aws서버에서 돌리려고 하였으나, 로컬에서 돌리는 시뮬레이터와 연결은 되지만 값 전달이 되지 않는 문제 발생

지나가는 값마다 port번호가 무작위로 생성되어 포트포워딩으로 불가

향후 개선 방향

VPN 이용

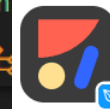
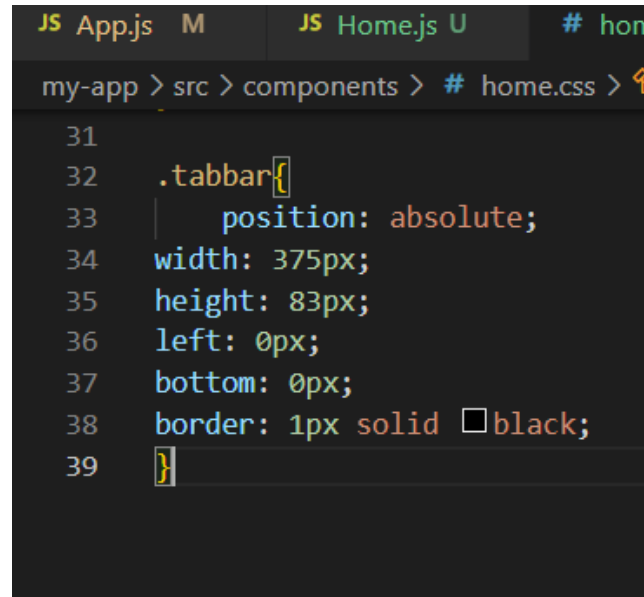
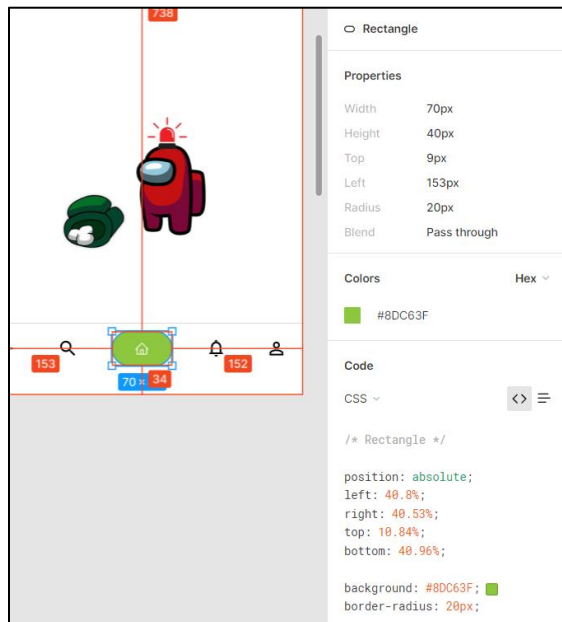
2

Front-End

2 Front-end

● Javascript+css 작성

이번주 목표 : figma to javascript + css



Anima - Export to React, HTML & Vue code

Accelerate the design to development process by getting code with Anima? Export Figma to HTML, CSS, React, or Vue



Locofy - Figma to React code (BETA)

Figma to React with Locofy Convert your Figma design to React code at lightning speed. ▶ Learn



Inspect - Export to HTML, React, TailwindCSS

Convert design to code with a click. No signups, no platform fees. Export to HTML or React with TailwindCSS. How to use it? Select a design



FigAct - Figma Designs to ReactJS

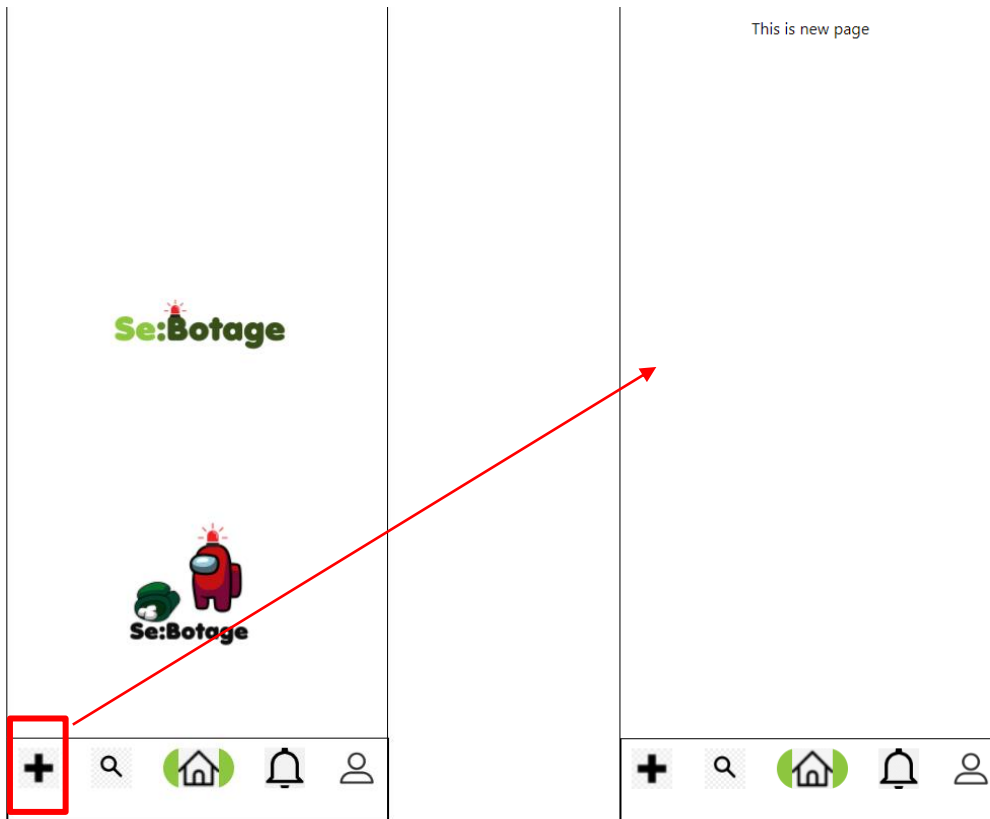
FigAct is a plugin to convert Figma designs into ReactJS code. The project is generated automatically by FigAct. Typing npm install figact

유료 플러그인 / 일정하지 않은 class name 지정 / html tag 배치의 문제

2 Front-end

● SPA

SPA 방식으로 페이지 레이아웃 제작



```
{article}  
<Navigation onChange={function(idx){  
  console.log('this is onChange function',idx)  
  if(idx=='plus'){  
    setArticle(<New />)  
  }  
  else if (idx=='search'){  
    setArticle(<Search />)  
  }  
}
```

Navigation component

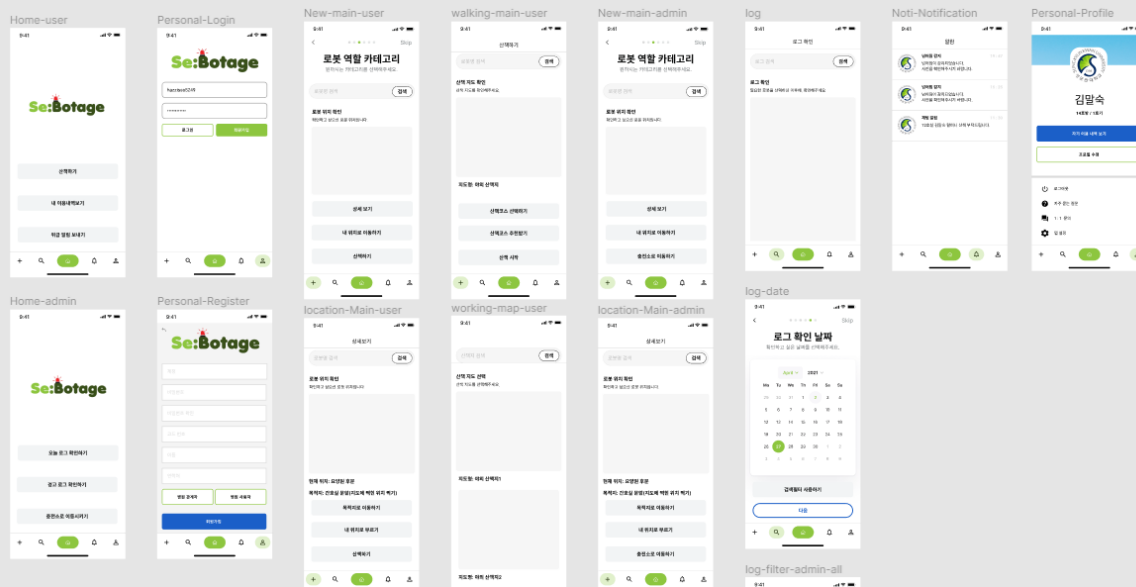
```
import React from 'react'  
  
function New() {  
  return (  
    <div>  
      <p>This is new page</p>  
    </div>  
  )  
}  
  
export default New;
```

New component

2 Front-end

다음주 목표

다음주 목표:
figma to javascript + css
로봇 상태확인 기능 구현



Figma pages into js, css

로봇 역할 카테고리

원하시는 카테고리를 선택해주세요.

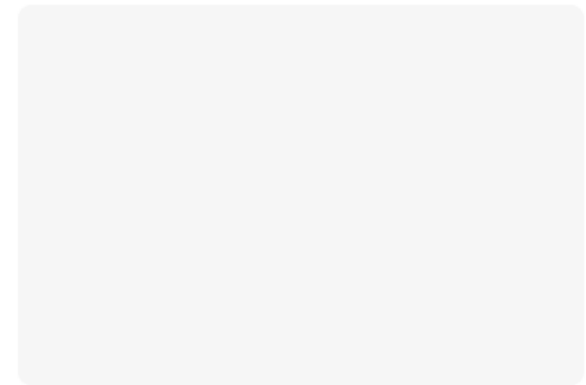
로봇명 검색

검색

로봇 상태 확인기능 구현

로봇 위치 확인

확인하고 싶으신 로봇 위치입니다.



상세 보기

내 위치로 이동하기

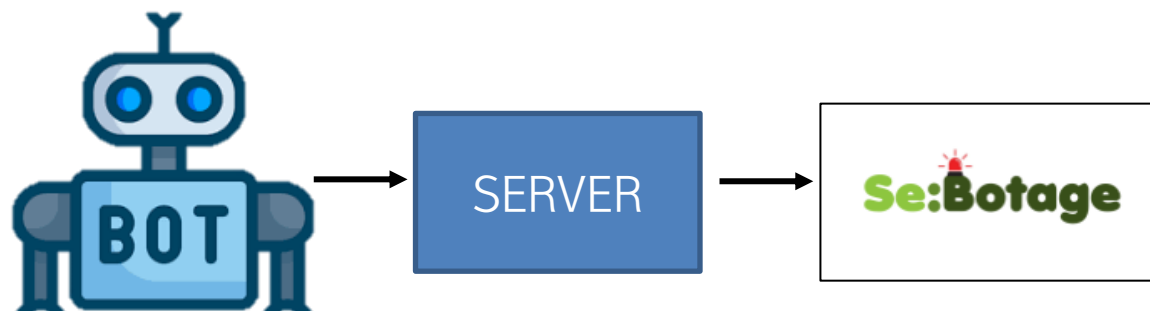
산책하기



2 Front-end

● 다음주 목표(2)

로봇이 현재 명령 수행중인지 여부
로봇의 현재 위치



전달방법

Websocket API

MQTT

ROSLIBJS

REST API ← 가장 빠르게 구현 가능

로봇 역할 카테고리

원하시는 카테고리를 선택해주세요.

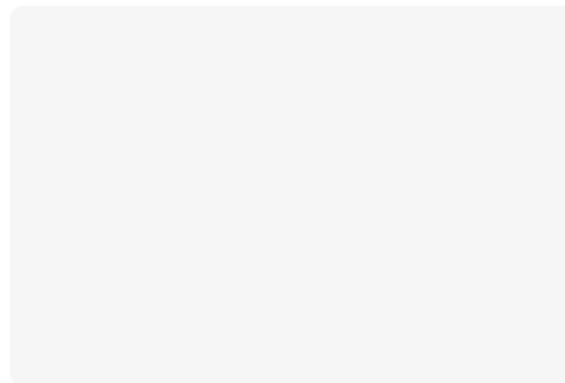
로봇명 검색

검색

로봇 상태 확인기능 구현

로봇 위치 확인

확인하고 싶으신 로봇 위치입니다.



상세 보기

내 위치로 이동하기

산책하기



3

Robotics_DL

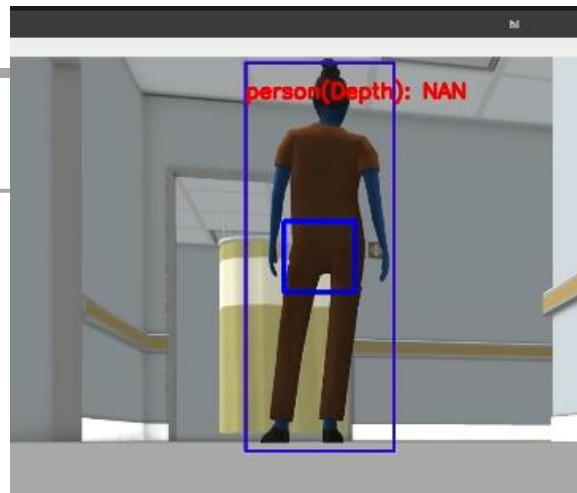
3 Robotics & DeepLearning

- 이번주 개선사항

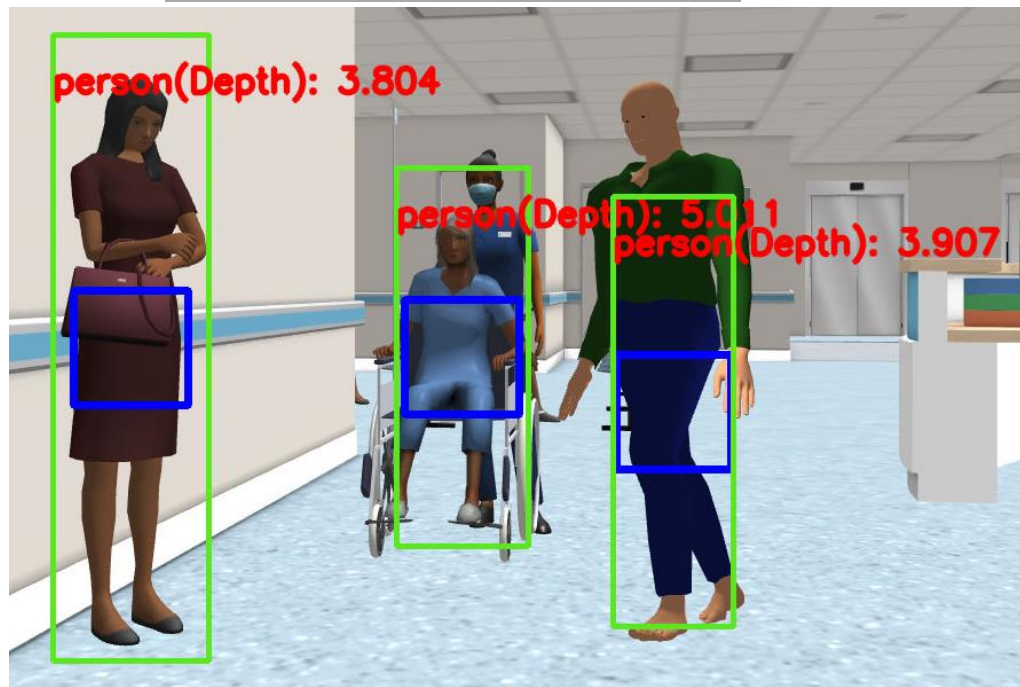
실시간 위치별 depth 정보 추출 시간 및 정확성 개선

기존 문제점 : - 가시거리에 비해 depth camera의 range가 적었다.
- 느려서 delay 발생하고, nan 값이 계속 발생하였다.

해결 방법 : - 기존 yolo.weight 파일 대신 tiny weigh파일을 써서 속도 개선 -> 정확도에도 큰 차이를 보이지 않음
- Depth camera의 max range를 수정함 5m -> 10m



Last Week



This Week

3 Robotics & DeepLearning

- Face recognition



목적: 사용자와 다른 사람들을 구별하기 위한 얼굴 인식

Source: face_recognition python api (https://github.com/ageitgey/face_recognition)

신경망: dlib 라이브러리 이용(face detection에 최적화된 cnn)

한계: - 느림 (real-time detect 불가) - 정확성 문제 (위 사진 참고)

3 Robotics & DeepLearning

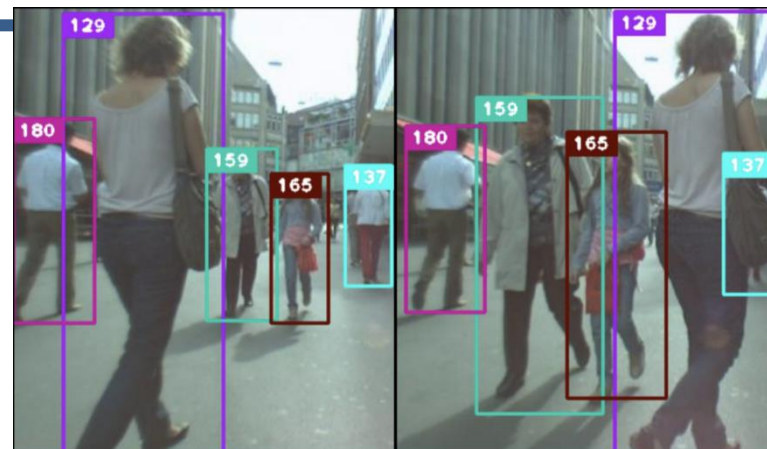
- 다음주 계획

사용자와의 위치를 고려한 산책 주행 알고리즘 개발

velocity command from Nav. Stack → Modify → Hardware

Sort tracking (object의 움직임을 추적)

Face recognition의 대안으로 사용자를 다른 사람으로부터 식별



4

Backend – login

4 Backend

● Register

RegisterDB

postgresql 사용해서 DB를 구현:
DB를 이전 항목에서 추가하였다.

idx [PK] bigint	user_id character varying (20)	user_pwd character varying (20)	user_name character varying (20)	user_code character varying (20)	user_number character varying (20)	user_type boolean	device_id character varying (50)	nursery boolean
4	skjunseo0103	!!save0506rl	김준서	2019313442	01092761606	true	[null]	false

```
CREATE TABLE IF NOT EXISTS public.member_info
(
    idx bigint NOT NULL DEFAULT nextval('member_idx_sequence'::regclass),
    user_id character varying(20) COLLATE pg_catalog."default" NOT NULL,
    user_pwd character varying(20) COLLATE pg_catalog."default" NOT NULL,
    user_name character varying(20) COLLATE pg_catalog."default" NOT NULL,
    user_code character varying(20) COLLATE pg_catalog."default" NOT NULL,
    user_number character varying(20) COLLATE pg_catalog."default",
    user_type boolean NOT NULL DEFAULT false,
    device_id character varying(50) COLLATE pg_catalog."default",
    nursery boolean NOT NULL DEFAULT false,
    CONSTRAINT member_pkey PRIMARY KEY (idx),
    CONSTRAINT member_info_user_code_key UNIQUE (user_code),
    CONSTRAINT member_info_user_id_key UNIQUE (user_id)
)
```

1. 다음과 같은 컬럼을 가지고 table을 형성함
2. 회원 번호와, 아이디만 unique 하게 설정
3. 이전에 db에서 device_id 그리고 nursery가 추가됨 : device_id는 알람을 위해서 사용하려고 함

4 Backend

- Register

Register Form

Register

회원 아이디

비밀번호

비밀번호 확인

회원명

회원 코드

전화번호

유형 ☐ 병원 관계자

1. 다음과 같은 register form을 가짐
2. 이전의 form에서 유형이 추가됨
3. 잘못해서 병원 관계자가 아닌데, 병원 관계자를 선택하여, 병원 관계자만이 접근할 수 있는 부분을 접근할 수 있다고 피드백 -> 수정 예정

4 Backend

- Login

Login Form

ID :

PWD :

이름 : 김준서 | [Edit](#) | [Logout](#)

ID :

PWD :

1. 이전과 형태는 동일함
2. 이전과 하나 달라진 점은 이제 ID에 회원 코드를 대신 적어도 로그인이 가능하게 만들
3. 로그인하면, 회원명을 session으로 보내주고 이를 통해서 로그인과 로그아웃을 구현

4 Backend

- Login

User_info_edit Form

회원정보 수정

User ID : 김준서

User Pwd :

User Code : 2019313442

수정

취소

1. 회원 정보 수정도 가능하게 함
2. 현재는 password와 usercode가 수정되도록 함
3. 정보 수정 버튼을 통해서 접근이 가능하게 할 예정이며, 추가적인 회원 정보 수정을 이야기를 통해서 구축할 예정



THANK YOU

