

# [이력서] JaeHwan

김재환 | JaeHwan  
Kim

Front-End Developer

🎂 1997.01.25

✉️ [ahhancom@gmail.com](mailto:ahhancom@gmail.com)

📌 [GitHub](#)

🌐 [포트폴리오 웹사이트](#)

📝 [노션 포트폴리오 보러가기](#)

## 소개 / About Me

- 4년 차 프론트엔드 개발자로 스타트업에서 웹서비스를 기획/개발합니다.  
가장 최근 근무 했던 '팀그릿'에서는 webRTC와 같은 기술을 사용하여 미디어 송수신 및 디바이스 컨트롤이 가능한 대화용 웹 서비스 제작을 하였습니다. 웹 서비스를 기한에 맞게 기획하고 개발하는 것에 대해 높은 역량을 가지고 있습니다.
- 사용해온 기술과 방식을 계속 의심하고 탐구하는 것을 좋아합니다. 의심하고 탐구하며 확신을 가지는 것에 매우 신중하기 때문에 다른 사람의 판단 혹은 결정도 쉽게 나오지 않는다고 생각합니다.
- 제가 경험한 프론트엔드 개발자는 디자이너, 백엔드 개발자, 디바이스 개발자 등 필연적으로 많은 사람과 협업을 하는 직무입니다. 따라서 협업의 중요성에 대해 깊이 이해하고, 더 나은 협업방식에 대해 고민하고 노력합니다.

## 경력 사항 / Work Experience 총 3년 4개월

재직 기간	회사명	부서 / 직급	담당 업무
2020.02.03 ~ 2021.04.13 (1년 3개월)	일삼공티	소프트웨어 개발자	- 블록 코딩 데스크탑 어플리케이션 개발

재직 기간	회사명	부서 / 직급	담당 업무
2021.05.10 ~ 2023.05.01 (2년 1개월)	팀그릿	프론트엔드 개발자	- 실시간 미디어 송수신 웹 개발 - 하드웨어 컨트롤

## 주요 프로젝트 / Projects

### 팀그릿 - FrontEnd

- 자체 미디어 서버를 가지고 초저지연 하드웨어 조종 플랫폼을 개발하는 회사입니다.

### 하드웨어 원격조종 플랫폼 웹 개발

2021.05.10 ~ 2023.05.01

#### 기술 스택

webBluetooth / Vue3 + pinia / TypeScript  
/ playwright / tailwindcss

#### 프로젝트 자세한 설명

Notion

- 백엔드, 하드웨어 개발자와 협업하여 로봇 등록 아키텍처 개발 후 로봇을 등록했을 경우 해당 로봇이 저장되어 로봇이 켜지는 즉시 원격 조종이 가능하도록 구현
- 실시간 미디어 송수신 최대 4개까지 할 수 있도록 개발
- 직접 송출 중인 미디어를 보고 선택하여 해당 미디어를 송출 중인 로봇을 컨트롤 할 수 있는 기능 개발
- 미디어 & 브라우저 최적화

#### 기술 도입

- 새로운 E2E 테스트 툴
  - playwright는 스크린테스트를 하지않고 한번에 여러개의 브라우저를 테스트할 수 있어 속도가 빠릅니다. Test generator를 사용하면 러닝커브가 많이 줄어듭니다. 새로운 팀원들도 쉽게 사용할 수 있고 속도도 빠르다고 판단하여 사용하게 되었습니다.

## 문제해결

- 멀티스레드 따라하기

- 4개의 미디어 중 하나라도 문제가 생겼을 때 웹이 멈추는 메인스레드 이슈를 해결하기 위해 webWorker를 사용하여 미디어는 전부 워커 스레드로 돌려서 미디어에 문제가 생기더라도 UX에는 문제없도록 개선하였습니다.

- 백엔드없이 백엔드와 협업

- msw는 기존의 코드를 전혀 수정하지 않아도 된다는 엄청난 장점을 가집니다. 덕분에 모킹서버를 사용 해야할 경우 코드를 변경해야하는 일이 없어졌고 api 혹은 네트워크에 문제가 있어도 프론트엔드팀은 아무 문제없이 개발을 진행 할 수 있었습니다.

- observer, singleton 패턴을 사용한 이벤트 관리

- 미디어, 블루투스, codec 등을 사용하기 쉽도록 만든 모듈들과 사용하는 페이지들이 점차 많아지고 api, ble 등의 타이밍 이슈와 worker를 사용함으로 인한 깊은 depth로 인해 이벤트나 상태 관리가 힘들어져 만들어진 것이 observer입니다. 무조건 하나의 인스턴스만 존재해야 하고 모든 곳에서 공유하기 위해 singleton으로 제작하고, 이벤트를 모두 이곳에서 관리하여 프로젝트의 이벤트 관리가 수월해졌습니다.

---

## 일본, 한국 원격 로봇 대전 웹 구축(Coplay 2022)

2022.01.01 ~ 2022.10.07

### 기술스택

webBluetooth / webCodec /  
Vue3 + vuex / TS / Cypress /  
tailwindcss / Amplify /  
graphql / 18n

### 프로젝트 자세한 설명

Notion

### 참고자료

### 주요 업무

- 일본에서 경기장 카메라를 송출하고 경기를 참여하는 한국과 일본의 참가자가 송출 중인 경기장으로 접속 후 경기장의 로봇을 조종하고 대결하는 웹을 제작
  - 실시간 미디어 송수신 페이지 개발.
  - 웹 블루투스를 사용하여 하드웨어와의 통신 개발.
  - 일본 대회 개최를 위한 현장 세팅, 네트워크 이슈 대응

- 대회를 성공적으로 마무리하여 시리즈A 투자 유치하는데 큰 역할을 했습니다.

## **기술 도입**

- **webpack vs Vite**
  - webpack의 devServer 구동 속도가 8초였습니다. 그래서 새로운 번들러를 찾았고 rollup과 Esbuild를 사용하는 vite를 사용하여 80%속도 증가를 이뤄냈습니다.
- **CSS 네이밍의 불편함과 반응형의 복잡함으로 인해 시작한 Tailwindcss**
  - 불필요한 네이밍을 하지 않고 반응형개발이 직관적이며 코드 길이가 짧아 tailwindcss를 사용하게 되었습니다.
- **UI 테스트의 낭비를 줄이기 위한 E2E**
  - 배포전 항상 직접 테스트하는 시간이 너무 길어 테스트를 만들게 되었습니다. 빠르게 만들어야 했기에 가장 보편적인 cypress를 사용하였고 테스트 개발 후 테스트 정확도 증가하고 배포 시간이 20분정도 줄었습니다.

## **문제해결**

- **미디어 사이즈 압축**
  - webRTC에서 회사의 솔루션으로 바뀌면서 websocket을 사용하게 되었고 jpeg 이미지로 영상 송수신을 하게 되었습니다. jpeg의 문제는 영상 송수신을 하기에는 높은 용량이었습니다. 그래서 webcodec을 사용하게 되었고 h264 codec으로 사이즈를 10배 이상 줄일 수 있었습니다.
- **mvvm 파이프라인 구축**
  - api를 이곳저곳에서 많이 사용하다 보니 관리가 힘들어져 api(model) → vuex(viewModel) → view 이 방식으로 구축하여 api의 데이터는 모두 vuex로 가고 view는 무조건 vuex에서 api 데이터를 받을 수 있도록 변경하였습니다. 변경 후에는 api 관련 문제가 생겼을 때나 api가 변경되었을 때 vuex를 수정하면 해당 api를 사용하는 모든 view가 변경되어 관리가 수월해졌습니다.

---

## **화성탐사를 컨셉으로한 WebRTC 레이싱 대회 웹 구축(Hello-Mars)**

2021.06.01 ~ 2021.11.10

## 기술 스택

Vue3 / SCSS / WebRTC / canvas

## 프로젝트 자세한 설명

Notion

## 참고자료

Youtube

WEB

NEWS

## 주요 업무

- 대회용 웹 구축
- 예약 서비스를 개발하여 대회가 시작하기 한 달 전에 유저들이 예약한 시간에 로봇을 조종하여 연습할 수 있도록 구현
- webRTC를 이용하여 실시간 미디어 통신 & 하드웨어 컨트롤 웹 개발
- 대회 중계 페이지를 제작하여 대회장에서 조종자들의 얼굴과 조종하고 있는 로봇의 카메라 그리고 대회장 전체를 볼 수 있는 중계 화면을 보는 페이지 구현
- 로봇 제작 참가자들을 위한 제작된 로봇을 조종하여 테스트 가능한 웹 구현

## 기술 도입

### • Vue2가 아닌 Vue3

- vue2가 많이 쓰여 라이브러리가 많고 vue3는 얼마안되어 라이브러리가 적어 많은 불편함이 존재했습니다. 하지만 typescript직접 지원(당장 사용하지는 않았지만)과 Composition API, 트리쉐이킹 강화, 가상 돔 오버헤드 감소 등 무시할 수 없는 변화에 vue3를 사용하게 되었습니다.

### • AWS 구축 : S3 + Route53 + CloudFront

- 회사자체에서 aws를 사용 하고는 있었지만 CSR웹은 따로 배포하지 않았고 워스정도만 사용하여 제가 스스로 배포방법을 찾아서 https로 배포해야 했습니다. 마침 혼자서 s3로 배포를 해보았던 경험이 있어서 직접 배포를 한다고 하였고 그렇게 cloudFront까지 사용하여 https로 배포하게 되었습니다.

## 문제해결

### • 시간과 태스크관리

- 대회가 대략 5개월 후였습니다. 3개월 후에는 예약 시스템을 제작해야 했고 그럼 2개월 만에 반응형 웹 제작과 미디어+하드웨어 컨트롤이 가능해야 했습니다. 우선순위 즉 빨

리 개발해야 하는 순서로 최소한의 시간을 들이기 위해 적은 시간이었지만 그 시간을 조각내고 태스크를 관리하여 프로젝트를 완수할 수 있었습니다.

- **미디어 권한 관리**

- 미디어 권한이 없는 유저들이 미디어 컨트롤 페이지에 바로 접속하는 경우 접속이 가능하여 미디어가 보이지 않는 현상이 생겨 Permissions API를 사용하여 미디어 권한이 없다면 접속을 금지하고 권한을 물어보도록 개선하였습니다.

- **image lazy loading**

- 웹을 처음 로딩 시 이미지마다 다운로드되는 속도가 달라 웹이 깨져 보이는 현상이 있었습니다. 해당 피드백을 들은 후 lazy loading을 사용하여 웹이 전부 다운로드 될 경우 이미지가 한 번에 로드되도록 개선하였습니다.

## 130T - Software Developer

- Scratch3와 자체제작 하드웨어를 접목하여 로봇 코딩교육을 하는 회사입니다.

### 코딩블록으로 하드웨어를 컨트롤하는 웹 개발(BlockBot)

2020.02.03 ~ 2021.04.13

#### 프로젝트 자세한 설명

Notion

#### 참고자료

download(window).

Wadiz

#### 주요 업무

- 코딩 교육용 앱인 스크래치 3를 기반으로 코딩 교육용 앱 개발
- DB를 사용하지 않고 JSON과 하드웨어에 데이터를 저장하여 DB처럼 사용
- React&Redux로 만들어진 스크래치 3를 개조하여 개발 진행. Electron을 이용하여 데스크톱 앱으로 배포.
- **결과**
- 와디즈 펀딩 성공.
- 초등학생을 타겟으로 해당 소프트웨어를 이용하여 교육 시행.

## 문제해결

- 첫 업무, 첫 개조 : 스크래치3 뜯어고치기
  - 팀에 합류하고 처음 맡게 된 프로젝트는 스크래치3 기반의 독자적인 소프트웨어를 만드는 것이었습니다. 0에서부터 하나씩 만들어 나가는 형태의 개발에 익숙했던 저는 완성된 프로그램을 개조하는 업무가 생소했고, 흔한 형태의 프로젝트는 아니었기에 참고할 만한 자료도 적어 진행에 어려움을 느꼈습니다.
  - 갓 입사한 신입 개발자였던 저는 플로우차트를 그려 스크래치3의 구조가 어떻게 동작하는지 파악해 전체 구조를 이해하고, 퇴근 후에는 스크래치3에 사용된 기술인 React&Redux와 JS 스터디를 통해 직무 전문성을 키워 정식으로 스크래치3 개조를 시작할 수 있었습니다.

## 기술 스택 / Skill Set

구분	skill
Framwork	Vue, React, Electron, Tailwindcss, PhaserJS
Languge	HTML/JS/CSS(Scss), TypeScript
CI	GithubAction
web Media	WebRTC, WebCodec
TEST	playwright, vitest, jest ,msw
AWS	Route53, CloudFront, S3, EC2
Package Manager	yarn berry(pnp)
API	REST, GraphQL (Apollo)

## 학력

- 2015 ~ 경남정보대학교 - 컴퓨터 정보 계열
- 2020.0302 ~ 2023 방송통신대학교 - 컴퓨터 과학과 졸업 예정

✉ [ahhancom@gmail.com](mailto:ahhancom@gmail.com)

📌 [GitHub](#)

🌐 [포트폴리오 웹사이트](#)