



김 성 욱

# Developer Portfolio

# 학습하는 # 프론트엔드

# 게임 클라이언트

# 디버깅 # React # C++

# 알고리즘 # 구현

# About Me

---

## Personality

상상하고 구현하는 것을 즐깁니다.

에러 발생 시 디버깅 하는 것을 좋아합니다.

꾸준한 학습을 통해서 빠르게 성장합니다.

## Experience

2020.03 ~ 2021.08 : 한양대학교 ICT건설기술연구실 학부 연구생

2022.05 ~ 2022.06 : 42Seoul La Piscine

2023.07 ~ 2024.06 : 삼성청년SW아카데미 10기

# About Me

---

## Contact

E-mail : bluebear9508@naver.com

Github : <https://github.com/Wookeykim95>

LinkedIn : <https://www.linkedin.com/in/sungwook-kim-b26aba228/>

## Education

2014.02 : 반포고등학교 졸업

2014.03 ~ 2018.09 : 연세대학교(원주) 물리학과

2019.03 ~ 2021.08 : 한양대학교 건설환경공학과

# Programming Language

---

C++

- 해당 언어를 이용한 기본 알고리즘 구현 가능  
(BFS, DFS, Dijkstra, MST, etc)
  - WinAPI를 사용하여 간단한 프로그램 구현 가능
  - 자료구조 관련 라이브러리 사용 가능  
(vector, string, map, queue, stack, priority\_queue, etc)
- 

JavaScript

- React를 사용하여 간단한 웹 페이지 구현 가능
  - Npm을 이용하여 라이브러리 관리 가능
- 

Python

- 해당 언어를 이용한 기본 알고리즘 구현 가능  
(BFS, DFS, etc)
  - 일부 라이브러리를 이용한 간단한 데이터 정리 가능  
(Matplotlib, Openpyxl, pandas, numpy)
- 

C#

- 작성된 코드에 대한 대략적인 이해 가능
- Unity C# Script 작성 가능

# Tools and Skills

Web FE	React		- 기본적인 코드 이해 가능 - 컴포넌트 구현 및 웹 페이지 구성 가능
	Linux		- 기본적인 명령어 사용 가능 - VIM을 통한 코드 작성 가능
	NextJS		- use client, use server에 대한 이해 가능 - 동적 라우팅을 위한 프로젝트 구조 이해 가능
Game Client	Unity		- C# 스크립트 작성을 통한 기능 구현 가능 - 화면 상 간단한 UI 구현 가능 - Scene 조작 및 오브젝트 배치 가능
	UE 4,5		- 블루 프린트를 통한 간단한 기능 구현 가능 - 오브젝트 배치 가능
Collaboration Tool	Git		- Github, Gitlab을 통한 협업 가능 - Pull, Merge, Push, Clone 등 명령어 사용 가능 - Git Convention에 대한 규칙 이해 및 준수
	JIRA		- 벤다운 차트 이해 가능 - Story Point, 백로그를 통한 업무 관리 가능

# Projects

## Chapter 1 FE



1. H.UP



2. Bon Voyage

## Chapter 2 Game Client

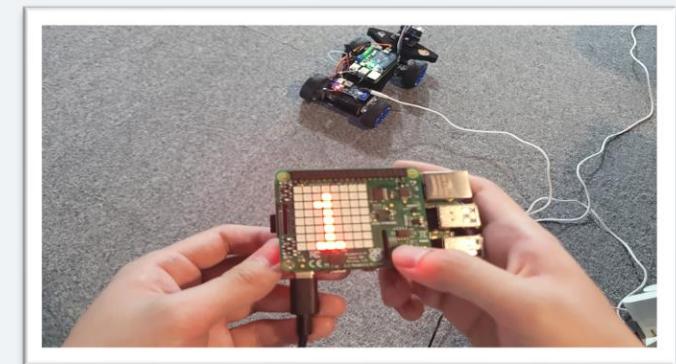


3. Michelin de 한양



4. CrazyArcade 모작

## Chapter 3 Algorithm



5. RC카 다중제어  
프로젝트



6. SSAFY RACE

# Project 1 - H.UP

개발기간 2024.04 ~ 2024.05 (6주)

개발인원 6명

역할 React 프론트엔드 개발

Yjs, BlockNote를 통한 동시편집 구현

Stomp, SockJS를 통한 웹소켓 연결

BlockNote 커스텀 블록 설정

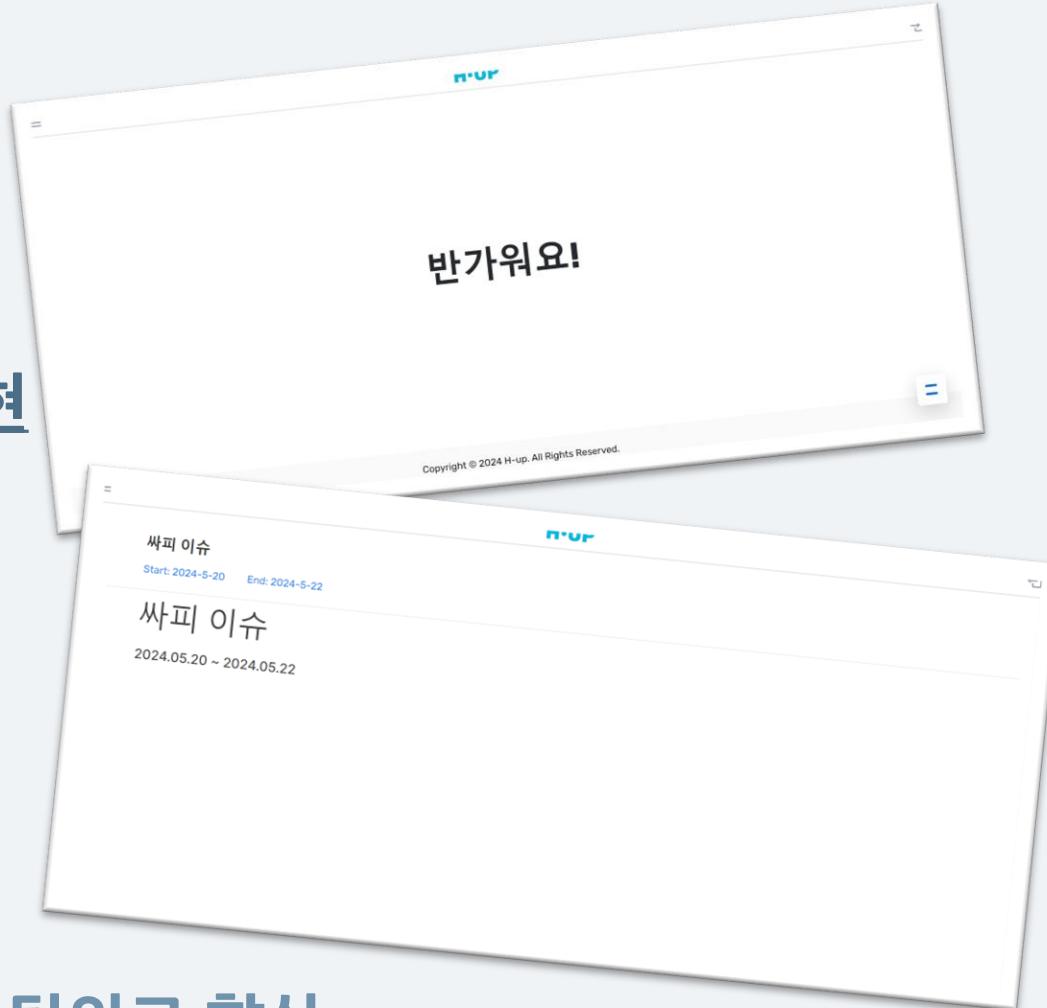
오류 발생 시 디버깅

성과 프론트 단 Socket 통신 구현

BlockNote를 통한 커스텀 블록 구현

Recoil을 통한 전역상태 관리 이해

Git, JIRA Convention 적용 및 실습, 팀워크 향상

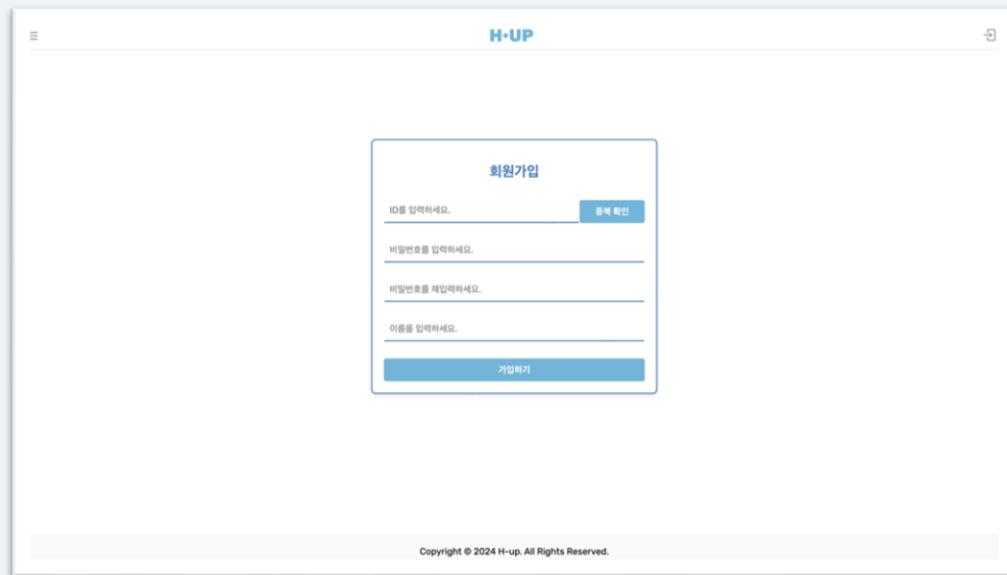


# Project 1 - H.UP

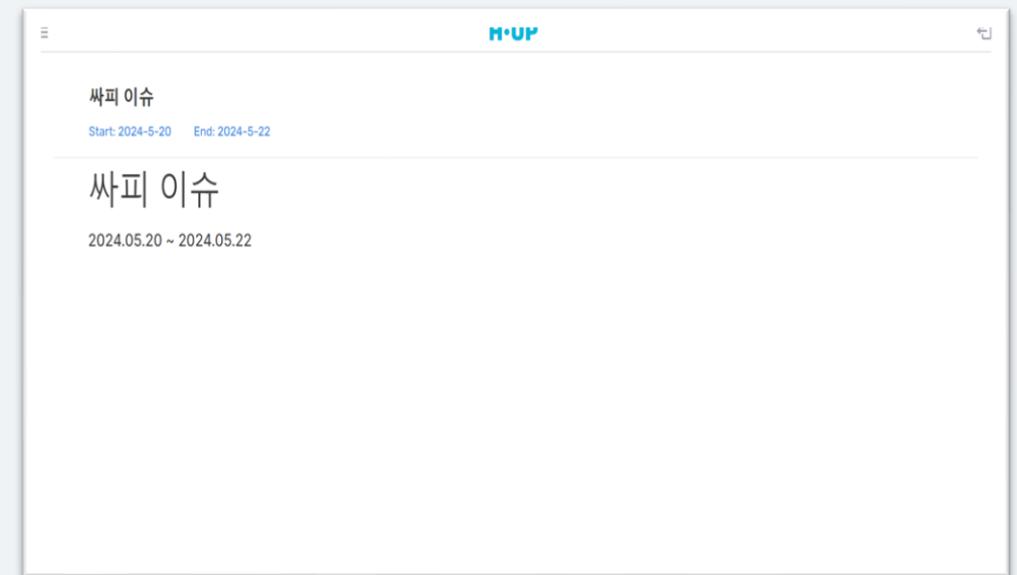
## 프로젝트 노션(ReadMe)

<https://tropical-manicure-529.notion.site/ReadMe-6a17a851a3444c3988d1ebf3e4411d54>

## 구현 내용



회원가입 페이지

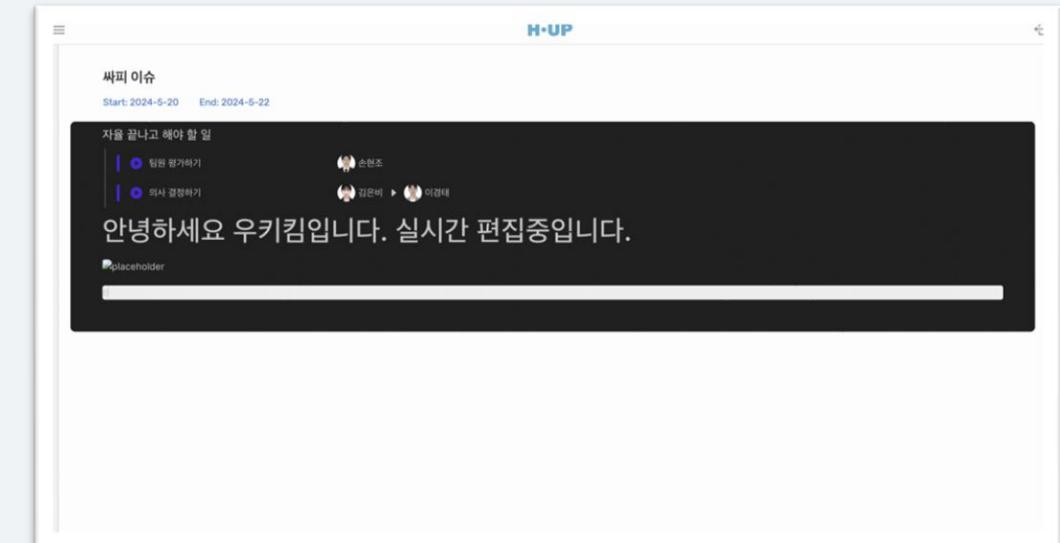


이슈작성 페이지

# Project 1 - H.UP

## 구현 내용

```
// ...  
  }  
  if (content && Array.isArray(content)) {  
    if (!pendingUpdates[messageId]) {  
      pendingUpdates[messageId] = new Array(totalChunks).fill(null);  
    }  
    pendingUpdates[messageId][chunkNum] = new Array(content);  
  
    if (!pendingUpdates[messageId].includes(null)) {  
      const concatenated = pendingUpdates[messageId].reduce(  
        (acc, val) => acc.concat(val),  
        []  
      );  
      const update = [];  
  
      for (let i = 0; i < totalChunks; i++) {  
        update.push(...concatenated[i]);  
      }  
  
      const newUpdate = new Uint8Array(update);  
  
      Y.applyUpdate(ydoc, newUpdate, 'remote');  
      delete pendingUpdates[messageId];  
    }  
  } else {  
    console.error('Invalid update format');  
  }  
} catch (error) {  
  console.error('Error applying updates:', error);  
}
```

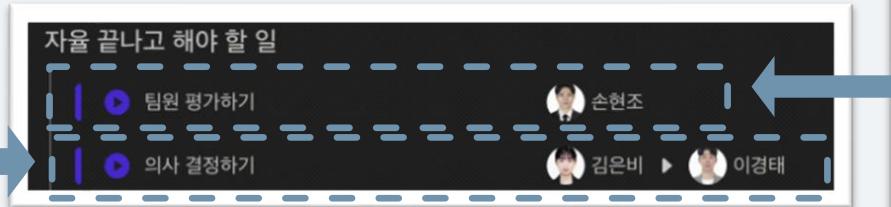


SockJS, Stomp를 이용한 Socket통신  
Yjs, BlockNote 상 실시간 동시편집

# Project 1 - H.UP

## 구현 내용

```
// The Agenda block.
export const Agenda = createReactBlockSpec(
{
  type: 'agenda',
  propSchema: {
    textAlign: defaultProps.textAlignment,
    textColor: defaultProps.textColor,
    type: {
      default: 'ASSIGNED',
      values: ['ASSIGNED', 'COMPLETED', 'COMPLETED', 'APPROVED'],
    },
    mention: {
      default: '엔센',
    },
    assignee: {
      default: '수신자',
    },
    assigneeimg: {
      default:
        'https://mblogthumb-phinf.pstatic.net/MjAyMDExMDFFMTgy/MDAxNjA0MjI4
    },
    mentionimg: {
      default:
        'https://mblogthumb-phinf.pstatic.net/MjAyMDExMDFFMTgy/MDAxNjA0MjI4
    },
    content: 'inline',
  }
},
```



```
// The Alert block.
export const Alert = createReactBlockSpec(
{
  type: 'alert',
  propSchema: {
    textAlign: defaultProps.textAlignment,
    textColor: defaultProps.textColor,
    type: {
      default: 'ASSIGNED',
      values: ['ASSIGNED', 'COMPLETED', 'COMPLETED', 'APPROVED'],
    },
    mention: {
      default: '엔센',
    },
    img: {
      default:
        'https://mblogthumb-phinf.pstatic.net/MjAyMDExMDFFMTgy/MD
    },
    content: 'inline',
  }
},
```

## BlockNote 커스텀 블록을 통한 Agenda, Todo 블록 구현

## 회고

실시간 동시 편집을 구현하면서 소켓 통신을 프론트단에서 구현해보는 시간을 가질 수 있었음.  
어떤 환경에서도 통신을 구현하고자 Yjs 통신이 아니라 Stomp와 SockJS를 사용하기로 하였음.  
CRDT를 적용하기 위해서 Uint8Array로 자료를 저장하고 ydoc업데이트를 적용하는 것이 어려웠지만 공식문서를  
찾고 GPT를 활용한 결과 Y.applyUpdate함수를 통해서 변경사항을 저장할 수 있었음.  
그리고 깃 컨벤션을 본격적으로 적용해보면서 팀워크를 향상시킬 수 있었던 시간이었음.  
또한, axios를 그냥 적용하지 않고, 하나의 파일에 모으고 필요할 때 각각 적용하는 방법에 대해서 배울 수 있었음.

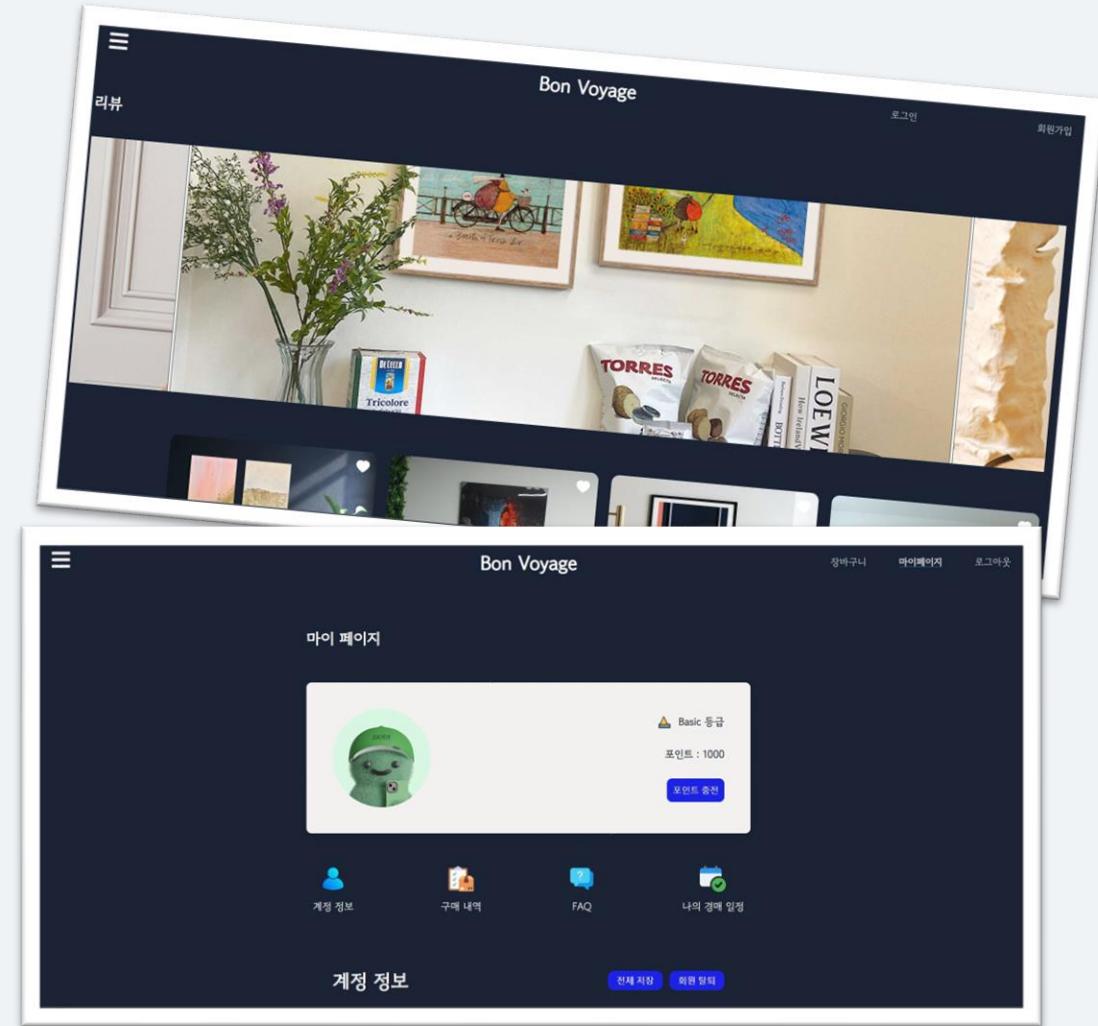
# Project 2 - Bon Voyage

개발기간 2024.01 ~ 2024.02 (7주)

개발인원 6명

역할 NextJS 프론트엔드 개발  
sessionStorage를 통한 로그인 구현  
오류 발생 시 디버깅

성과 NextJS 기초에 대한 이해  
use client, use server 구분 가능  
SSR, 동적 라우팅에 대한 이해

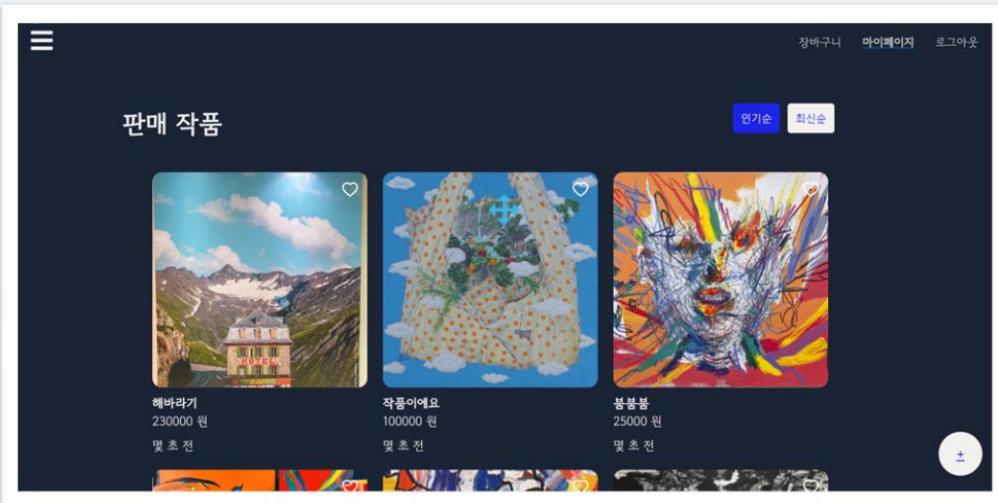


# Project 2 - Bon Voyage

## 레포지토리

<https://github.com/WookeyKim95/BonVoyage>

## 구현 내용



판매작품 페이지

The screenshot shows a registration form titled '개인 회원가입' (Personal Membership Registration). The form contains the following fields:

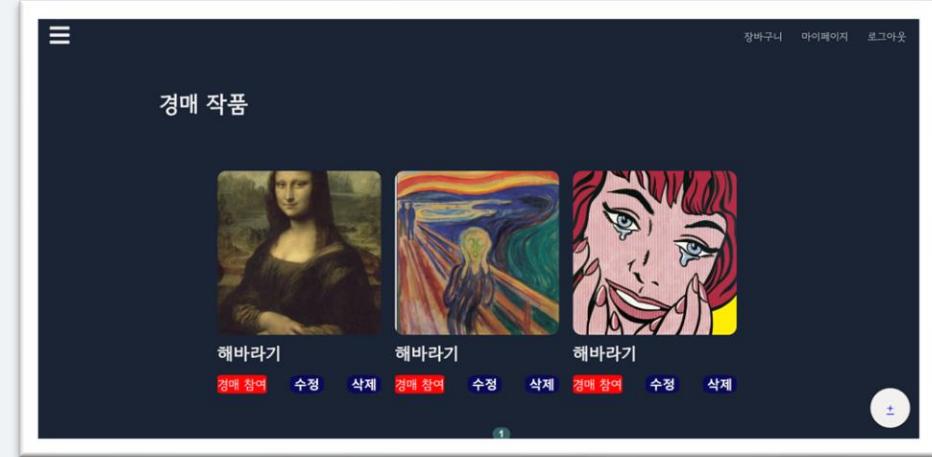
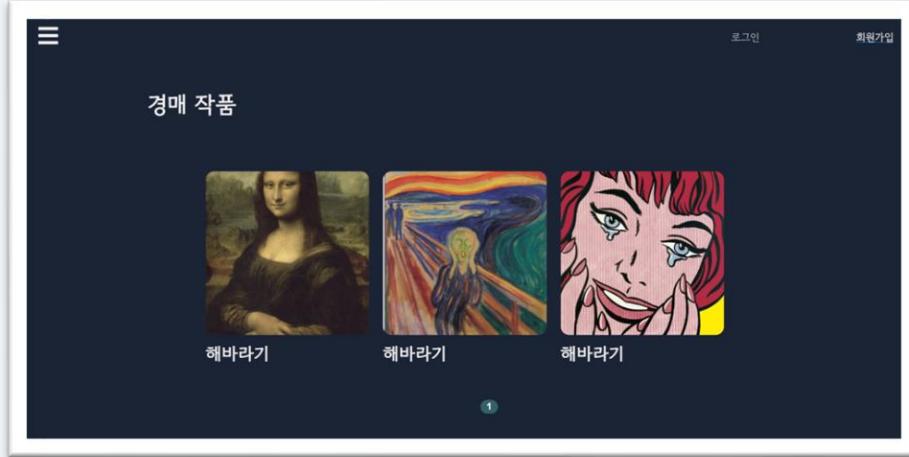
이름	이름
아이디	아이디
비밀번호	비밀번호
비밀번호 확인	비밀번호 확인
닉네임	닉네임
이메일	이메일
연락처	연락처

At the bottom right of the form is a blue button labeled '인증하기' (Verify).

회원가입 페이지

# Project 2 - Bon Voyage

## 구현 내용



경매작품 페이지, 로그인 세션 상태에 따라 버튼이 나타나게 구현

## 회고

프론트엔드 첫 프로젝트다 보니 리액트 사용법에 익숙치 않았던 점이 아쉬웠음.

지식이 부족한 상태에서 무조건 NextJS가 좋은 기술인줄 알고 적용을 해버린 것이 실수였다고 생각함.

만들고 나니 서비스 자체가 SSR보다는 CSR을 사용하는 것이 더 적합하겠다고 생각했음.

그 이유는 메인 페이지에서의 Three.js 사용이었고, 용량이 큰 페이지를 서버보다는 클라이언트에서 페이지를 처음 접속한 시점에 렌더링 하는 것이 서버 부담을 줄여줄 것이기 때문임.

아쉬움이 많았지만 기술 스택을 사용할 땐 그 이유를 따져야 한다는 것을 배울 수 있었음.

# Project 3 - Michelin de 한양

개발기간 2024.02 ~ 2024.04 (7주)

개발인원 6명

역할 팀장

Unity 클라이언트 UI 개발

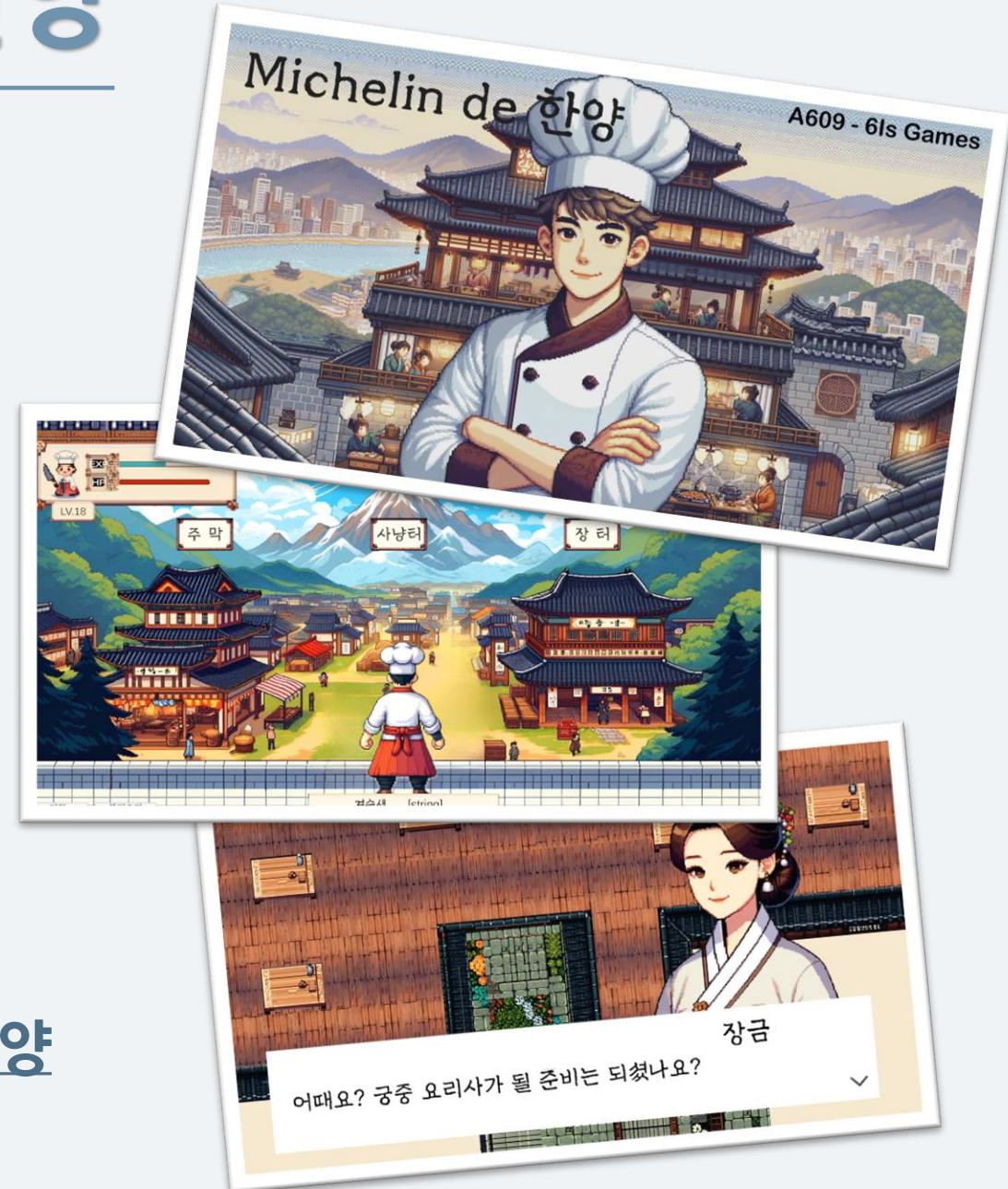
UCC 편집, 발표

스토리 및 대사 구성, 에셋 제작

성과 특화 프로젝트 우수상 수상

Unity 클라이언트 개발 능력 함양

일정 관리, 역할 분담 등 리더십 함양



# Project 3 - Michelin de 한양

## 레포지토리

<https://github.com/WookeyKim95/Michelin-de-Hanyang>

## 구현 내용



대화 창 UI 구성



화면 좌측 차림판 UI 구성

# Project 3 - Michelin de 한양

## 구현 내용

```
public bool CanIncrease(ref MenuItem menuItem) {
    // 재료 체크 로직

    if (menuItem.name == "밥" || menuItem.name == "김치" || menuItem.name == "도토리묵") {
        if (menuItem.currentAmount < 3) {
            return true;
        }
        else {
            return false;
        }
    }

    if (ingredients.Count == 0) {
        return false;
    }
    foreach (KeyValuePair<Ingredient, int> ingredient in menuItem.NeedIngredient) {
        bool flag = false;
        // Debug.Log(ingredient.Key.itemcode);
        for (int i = 0; i < ingredients.Count; ++i) {
            if (ingredient.Key.itemcode == ingredients[i].itemcode) {
                if (ingredient.Value <= ingredients[i].amount) {
                    flag = true;
                }
            }
        }
        if (!flag) {
            return false;
        }
    }
}
```

```
for (int i = 0; i < menuManager.SoldItems.Count; ++i) {
    questComplete.SoldFood.Add(menuManager.SoldItems[i].name, menuManager.SoldItems[i].currentAmount);

    bool flag = false;
    for (int j = 0; j < GameMgr.Instance.soldFood.Count; ++j) {
        if (GameMgr.Instance.soldFood[j].code == menuManager.SoldItems[i].itemCode) {
            GameMgr.Instance.soldFood[j].count++;
            flag = true;
            break;
        }
    }

    if (!flag) {
        LoginManager.Food newItem = new LoginManager.Food();
        newItem.code = menuManager.SoldItems[i].itemCode;
        newItem.count = 1;
        GameMgr.Instance.soldFood.Add(newItem);
    }

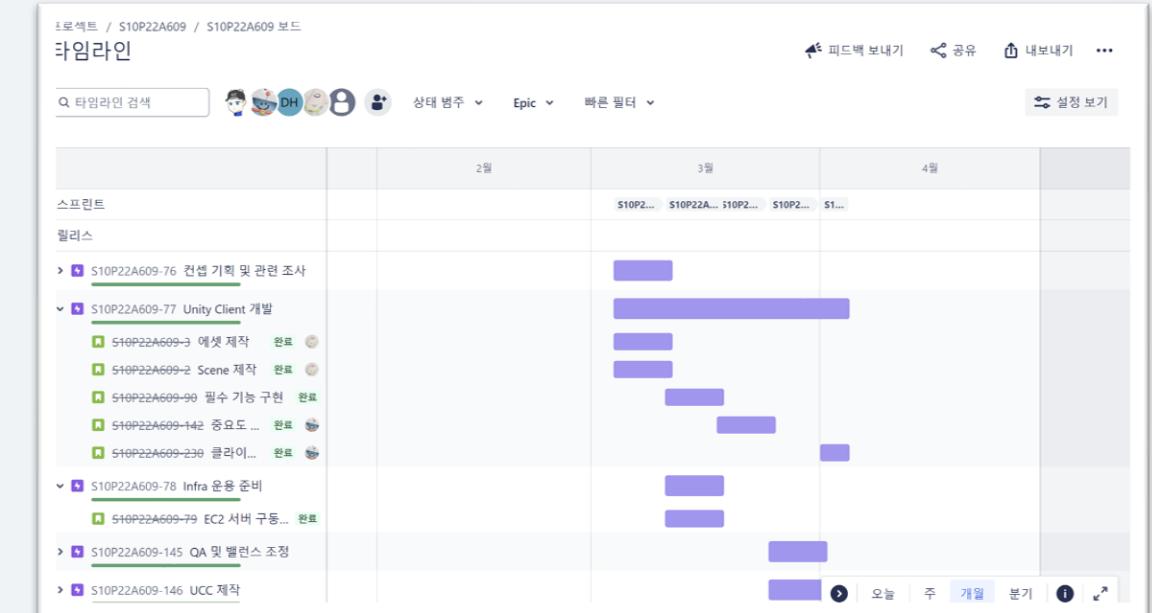
    ResultContent += menuManager.SoldItems[i].name + ' ' + menuManager.SoldItems[i].currentAmount.ToString() + '₩' + '\n';
    ResultCash += menuManager.SoldItems[i].MenuCost * menuManager.SoldItems[i].currentAmount;
}
```

## 영업 종료 후 상품 판매 정산 로직 설계

## 영업 전 메뉴 수량 선택 및 제한 로직 설계

# Project 3 - Michelin de 한양

## 구현 내용



반다운 차트와 타임라인을 통한 일정 관리

# Project 3 - Michelin de 한양

## 구현 내용



ChatGPT4, DALL-E를 사용한 배경 및 에셋 제작

## 회고

처음으로 팀장을 맡았음. 팀장으로서 팀원의 경험을 대화를 통해서 알아보니 게임 클라이언트 개발을 나보다 더 잘하는 사람이 두명이나 있었음. 특히 한 분은 이미 유니티 유경험자였음.

그래서 UI, 스프라이트, 에셋과 간단한 로직으로 설계할 수 있는 기술을 맡고, 조금 복잡한 로직을 구성할 필요가 있는 기술의 구현은 그 두분에게 맡기고 그들이 최대한 개발에 집중할 수 있도록 지원하고자 했음. 그리고 JIRA를 통해서 일정관리를 했고, 그 결과 기한 내 구현 목표를 달성하고 특화프로젝트 우수상을 수상함. 더 깊은 기술을 고민할 기회가 부족했던 것은 아쉽지만 팀장으로서는 성공적인 경험이었다고 생각이 들었음.

# Project 4 - CrazyArcade 모작

개발기간 2022.09 ~ 2022.10 (4주)

개발인원 1명 (개인 프로젝트)

역할 C++/WinAPI를 이용한 게임 개발  
Asset, Sprite 제작

성과 WinAPI를 활용한 프로그래밍 이해  
게임 엔진의 기본 구조 이해  
소개 영상 링크 : [https://youtu.be/EJj\\_PRnWfNE](https://youtu.be/EJj_PRnWfNE)

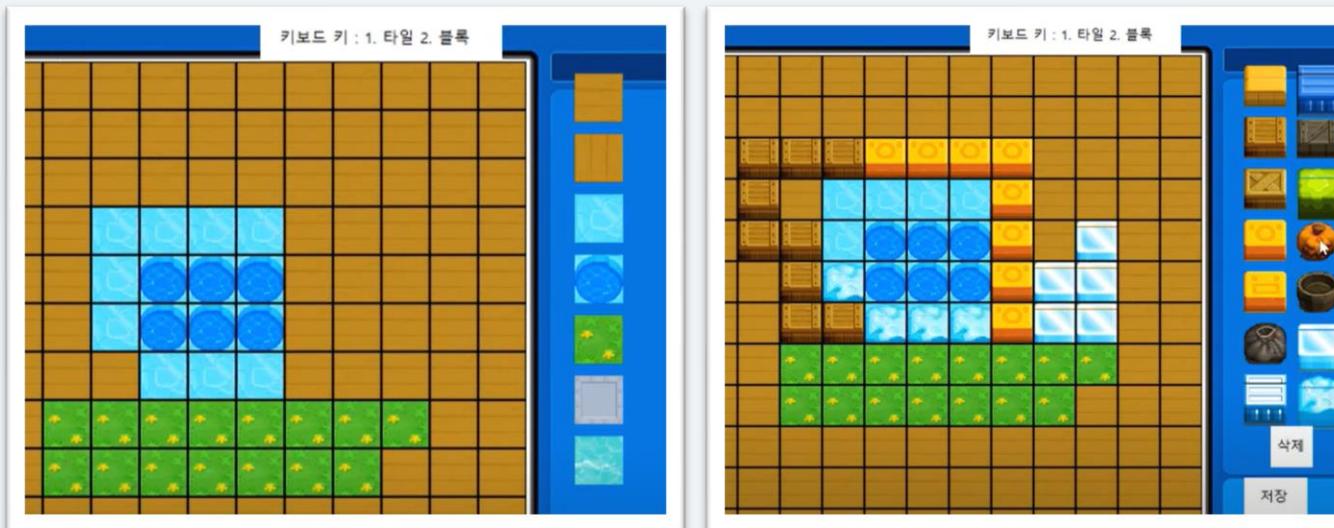


# Project 4 - CrazyArcade 모작

## 레포지토리

<https://github.com/WookeyKim95/CrazyArcade>

## 구현 내용

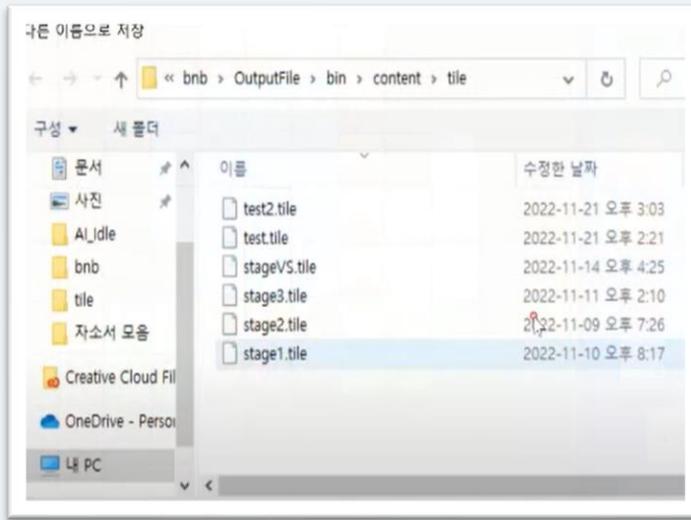


타일 맵 에디터

- 1번을 누를 경우 바닥 타일 팔레트를 불러옴.
- 2번을 누를 경우 블록을 팔레트를 불러오도록 구현.
- 팔레트에 있는 블록이나 타일을 클릭해서 원하는 부분에 좌클릭해서 맵을 구성할 수 있음.

# Project 4 - CrazyArcade 모작

# 구현 내용



```
void CEditorLevel::SaveTitle()
{
    // open a file name
    OPENFILENAME ofn = {};

    wstring strTitleFolderPath = CPathMgr::GetInst()->GetContentPath();
    strTitleFolderPath += L"titleWW";

    ...

    wchar_t szFilePath[256] = {};
    ZeroMemory(&ofn, sizeof(ofn));
    ofn.lStructSize = sizeof(ofn);
    ofn.hwndOwner = NULL;
    ofn.lpstrFile = szFilePath;
    ofn.lpstrFile[0] = 'W';
    ofn.nMaxFile = 256;
    ofn.lpstrFilter = L"ALLWO.*WOTileWO*.tileWO";
    ofn.nFilterIndex = 1;
    ofn.lpstrFileTitle = NULL;
    ofn.nMaxFileTitle = 0;
    ofn.lpstrInitialDir = strTitleFolderPath.c_str();
    ofn.Flags = OFN_PATHMUSTEXIST | OFN_FILEMUSTEXIST;
```

```

// 각각의 타일 정보 저장
const vector<CObj*>& vecTile = GetLayer(_layer: LAYER::TILE);
for (size_t i = 0; i < vecTile.size(); ++i)
{
    ((CTile*)vecTile[i])->Save(_pFile: pFile);

}

// 각각의 박스(파괴가능) 정보 저장
const vector<CObj*>& vecBox = GetLayer(_layer: LAYER::BOX_BREAK);
for (size_t i = 0; i < vecBox.size(); ++i)
{
    ((CBox_Break*)vecBox[i])->Save(_pFile: pFile);

}

// 각각의 장애물여부 정보 저장
const vector<CObj*>& vecObstacle = GetLayer(_layer: LAYER::OBSTACLE);
for (size_t i = 0; i < vecObstacle.size(); ++i)
{
    ((CObstacle*)vecObstacle[i])->Save(_pFile: pFile);
}

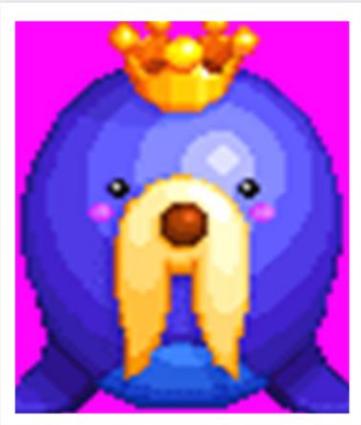
```

.tile 형태로 파일 형식을 구성하였으며, 아래의 정보를 순차적으로 저장하고 불러오도록 구현함.

- 타일의 좌표
  - 텍스쳐
  - 장애물 여부
  - 파괴 가능 여부

# Project 4 - CrazyArcade 모작

## 구현 내용



```
void CSeal::BeginOverlap(CCollider* _pOther)
{
    Vec2 vPos = GetPos();

    if (_pOther->GetOwner()->ReturnLayer() == LAYER::WAVE || _pOther->GetOwner()->ReturnLayer() == LAYER::WAVE)
    {
        if (!m_bHit && m_HP > 0)
        {
            --m_HP;
            m_bHit = true;
            GetMonsterAI()->ChangeState(_strStateName: L"Seal_Hitted");
        }

        if (m_HP == 0)
        {
            GetMonsterAI()->ChangeState(_strStateName: L"Seal_Trapped");
            m_BTrapped = true;
        }
    }
}
```

몬스터와 물줄기가 닿을 경우

데미지를 입거나,

사망모션을 재생하도록 구현

# Project 4 - CrazyArcade 모작

## 구현 내용

```
// 상하좌우 키로 움직이기  
if (IsPressed(KEY::RIGHT) && !(IsPressed(KEY::DOWN)) && !(IsPressed(KEY::UP)) &&  
    vPos.x += pPlayer->m_fSpeed * DT;  
pPlayer->SetPos(_v: vPos);  
  
if (IsPressed(KEY::LEFT) && !(IsPressed(KEY::DOWN)) && !(IsPressed(KEY::UP)) &&  
    vPos.x -= pPlayer->m_fSpeed * DT;  
pPlayer->SetPos(_v: vPos);  
  
if (IsPressed(KEY::DOWN) && !(IsPressed(KEY::UP)) && !(IsPressed(KEY::RIGHT)) &&  
    vPos.y += pPlayer->m_fSpeed * DT;  
pPlayer->SetPos(_v: vPos);  
  
if (IsPressed(KEY::UP) && !(IsPressed(KEY::RIGHT)) && !(IsPressed(KEY::DOWN)) &&  
    vPos.y -= pPlayer->m_fSpeed * DT;  
pPlayer->SetPos(_v: vPos);
```

상하좌우 키를 누를 시  
해당 방향으로 이동



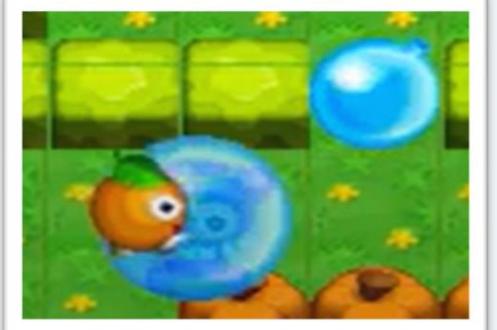
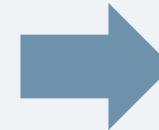
```
if (Islap(KEY::SPACE) && pPlayer->m_BombCount > 0 && pPlayer->m_CanSetBomb)  
{  
    // 물폭탄 생성  
    CLevel* pCurLevel = CLevelMgr::GetInst()->GetCurLevel();  
  
    CBomb* pBomb = new CBomb(pPlayer, pPlayer->GetWaveCount()); // 이 플레이어가  
    // 물풍선 좌표 설정  
    pBomb->SetScale(_v: Vec2(40.f, 40.f));  
    Instantiate(_pNewObj: pBomb, _vPos: pPlayer->m_TileCenter, _eLayer: LAYER::BOMB);  
    --(pPlayer->m_BombCount);  
    pPlayer->m_BombFirst = true;  
}
```

SPACE를 누를 시  
- 물풍선이 놓일 좌표 계산  
- 물풍선을 타일 중앙에 생성.

# Project 4 - CrazyArcade 모작

## 구현 내용

```
else if ((_pOther->GetOwner()->ReturnLayer() == LAYER::WAVE || _pOther->GetOwner()->ReturnLayer() == LAYER:  
|| _pOther->GetOwner()->ReturnLayer() == LAYER::MONSTER_OBJECTILE && !_m_BTrapped)  
&& !_m_BTrapped && !_m_PowerOverwhelming)  
{  
    ChangeState(_strStateName: L"Trapped");  
    m_BTrapped = true;  
}
```



Trapped 상태가 아닌 조건에 물줄기나 몬스터와 접촉할 시  
물방울에 갇힌 상태 즉, Trapped 상태로 변환함.

## 회고

물풍선이 터질 때 나타나는 한 칸의 물줄기도 하나의 객체임. 그런데 물풍선이 많이 터지는 만큼 물줄기가 많이 나타나면 객체가 많이 생성되면서 프레임 드랍이 일어나는 문제가 있었음.

이를 복사 생성자를 통해 해결하였음. 복사 생성자를 통해서 이미지 파일을 로딩하는 과정을 생략해서 프레임 드랍을 약 95% 줄일 수 있었음.

그 당시에는 ChatGPT가 아직 활성화 되지 않은 시기였기 때문에 힘들었지만 검색 역량과 복습에 대한 의지를 향상시킬 수 있었음.

# Project 5 - RC카 다중제어 프로젝트

개발기간 2023.11 (5일)

개발인원 4명

역할 AWS 네트워크에서 수신한  
데이터 파싱

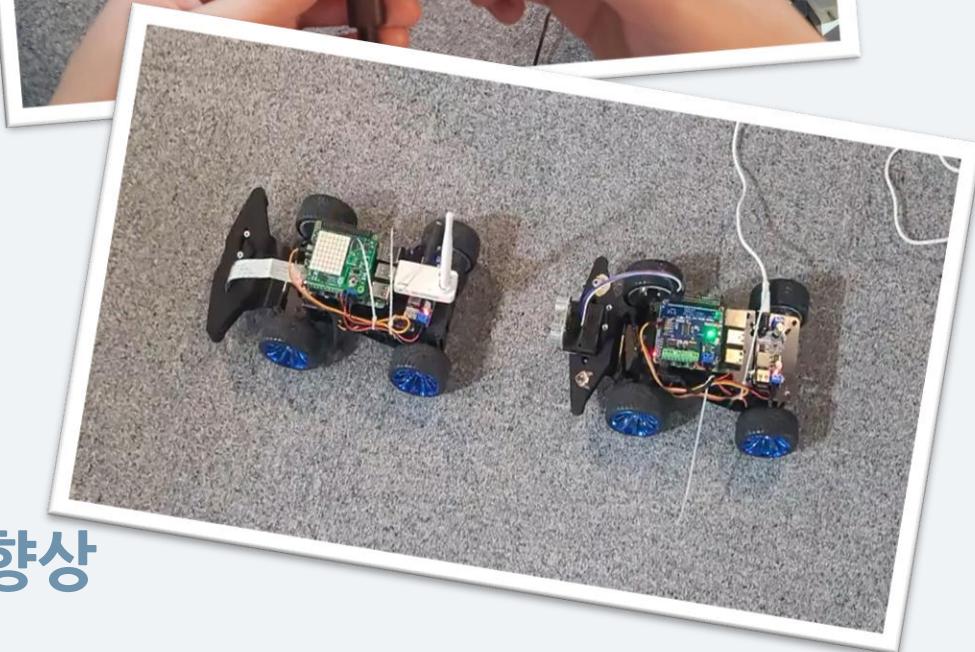
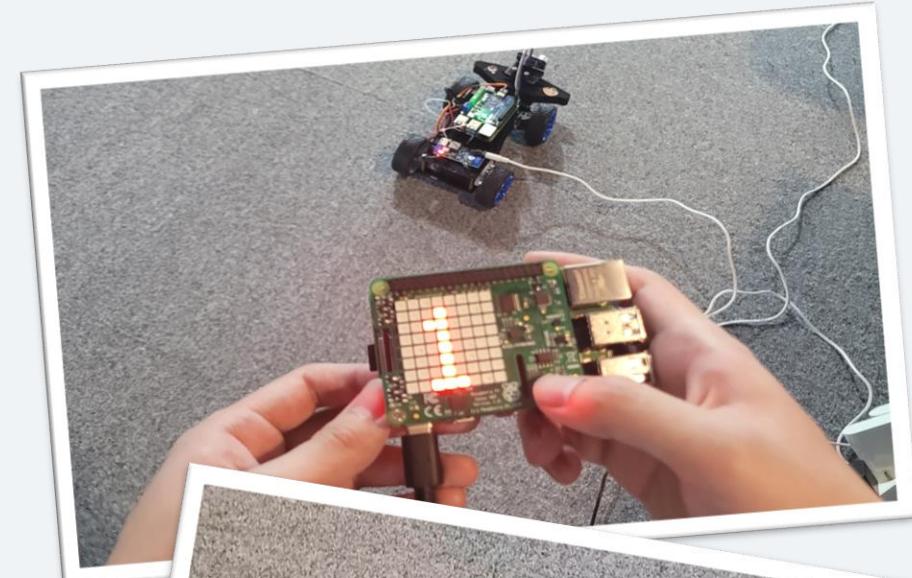
RC카 제어 로직 설계

RC카 제어 파라미터 조정

성과 구독, 발행에 대한 개념 이해

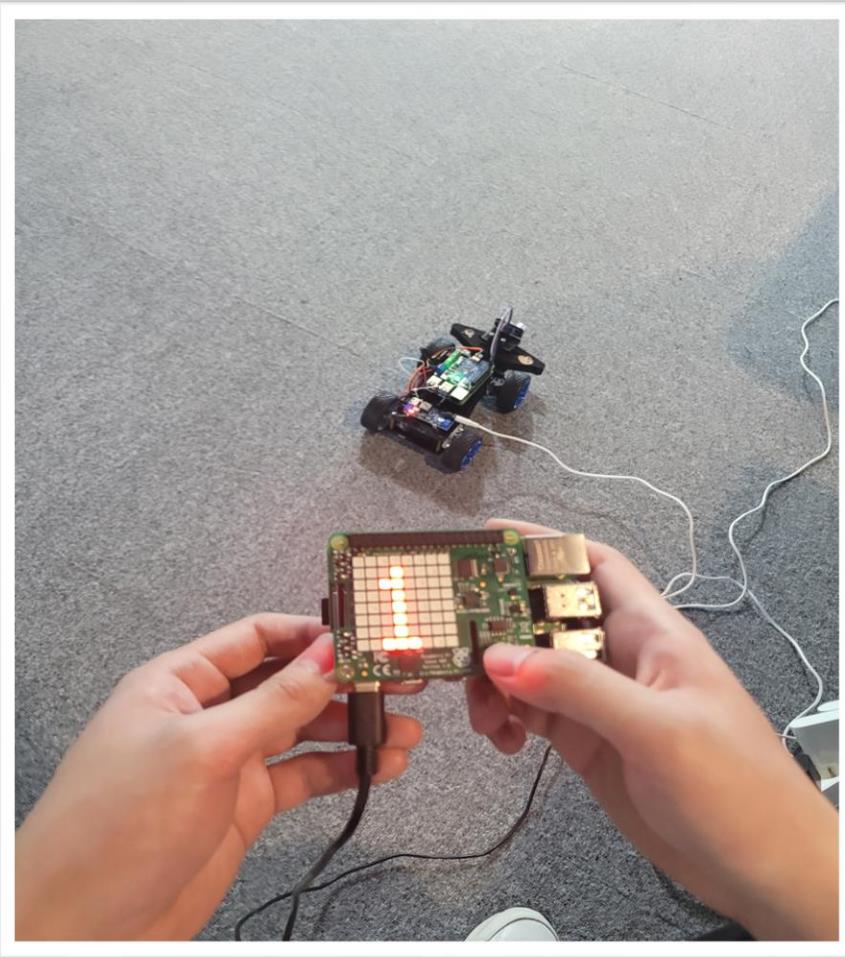
Linux 환경 개발 능력 향상

라즈베리파이 환경 Python 개발 능력 향상



# Project 5 - RC카 다중제어 프로젝트

## 구현 내용



SenseHat 을 이용해  
조이스틱 구성

조이스틱의 입력 값을  
AWS를 통해 전송

# Project 5 - RC카 다중제어 프로젝트

## 구현 내용

```
# Callback when the subscribed topic receives a message
def on_message_received(topic, payload, dup, qos, retain, **kwargs):
    # 0:F 1:B 2:L 3:R
    #print("Received message from topic '{}' : {}".format(topic, payload))

    # convert bytes type to string(json
    msg = payload.decode('utf-8')

    # convert json to dict
    msg = json.loads(msg)
    # print(f"decoded string : {msg}, {type(msg)}")

    # get message
    message = msg
    sign = message["message"]
    global received_count
    received_count += 1

    print(sign)
```

조이 스틱 조작 시  
메시지 수신

```
if sign == 1:
    Speed += Accel
elif sign == 0:
    Speed -= Accel
elif sign == 2:
    Direction += 50
elif sign == 3:
    Direction += 50

if Speed > 255:
    Speed = 255

if Speed < -255:
    Speed = -255

if Direction >= 400:
    Direction = 400
if Direction <= 200:
    Direction = 200

FB = 1
if Speed < 0:
    FB = -1

if FB == 1:
    myMotor.run(Raspi_MotorHAT.FORWARD)
elif FB == -1:
    myMotor.run(Raspi_MotorHAT.BACKWARD)

myMotor.setSpeed(Speed * FB)
servo_control.setPWM(0, 0, Direction)
```

메시지 값에 따라  
속도, 방향 전환

입력한 값을  
하드웨어에 전송

## 회고

파라미터를 조정하고 제어 로직을 설계하는 과정을 맡았고, SSAFY RACE 프로젝트와 더불어 시뮬레이션과 알고리즘, 연구에 대한 열정이 나타나는 프로젝트였다고 생각함.

더불어, Makefile을 만드는 등 쉘 스크립트 코딩을 해봄으로써 Linux 환경 아래에서 코딩 능력을 향상 시킬 수 있는 프로젝트였다고 생각함.

# Project 6 - SSAFY RACE

개발기간 2023.12 (4주)

개발인원 3명

역할 도로분할 알고리즘 설계

장애물 회피 전략 및 알고리즘 설계

파라미터 조정

성과 Speed 맵 전국 1등, SSAFY 맵 전국 2등

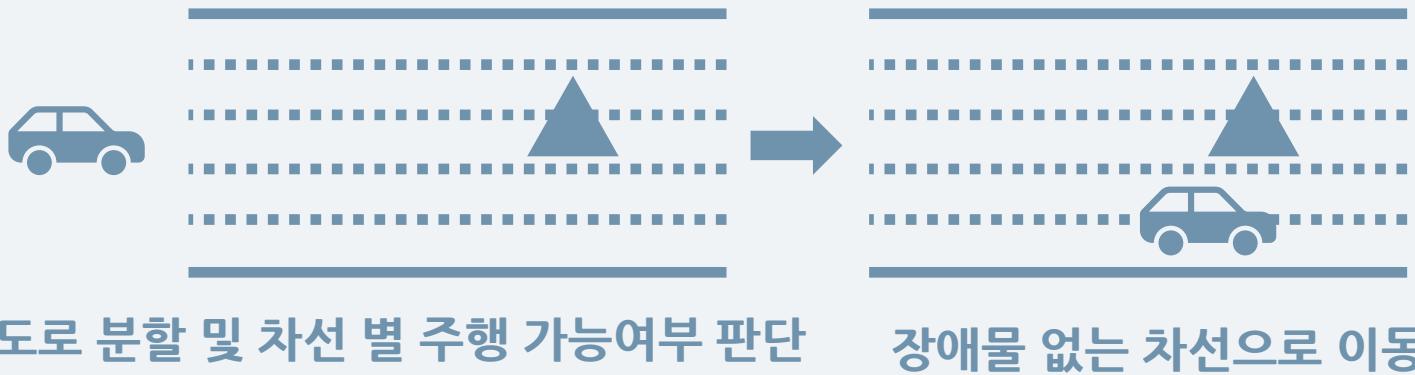
알고리즘 설계 능력 향상

역할 분담 등 팀워크 능력 향상



# Project 6 - SSAFY RACE

## 구현 내용



## 회고

도로를 나누자는 조원의 아이디어를 바탕으로 도로 분할 알고리즘 및 장애물 탐지 알고리즘을 설계하였음.  
분할한 도로 수가 적었기 때문에 완전탐색 알고리즘을 사용하였으나 더 빠른 탐색 알고리즘을 이용할 수 있는 방법을 찾지 못한 것이 아쉬웠음.

대신에 주행 위치를 정하는데 용이하게 하기 위해서 2, 4, 8, 16 등 2의 급수 개수 만큼 나누도록 설계함.  
밤을 새면서 시뮬레이션을 돌리고 파라미터를 조정하는 모습을 통해서 알고리즘 설계와 코딩에 대한 열정을 스스로 확인할 수 있었고, 결실을 맺은 프로젝트였음.

## 주행 전략

- 전방 장애물 개수에 따라 최대속도 제한  
(회피를 쉽게 하기 위함)
- 장애물이 없을 시에는 최대 가속

**Thank You!**