



# 김대호

010-8978-5036

[aquhm@naver.com](mailto:aquhm@naver.com)

<https://github.com/aquhm>

한국공학대학교

게임공학과 (2002 - 2010)

## ABOUT ME

안녕하세요. 개발자 김대호입니다.

읽기 쉬운 코드를 작성하는 것을 지향하며, 협업의 중요성을 깊이 인식하고 있습니다. 새로운 시스템의 아키텍처를 설계할 때, 팀원들이 빠르게 익숙해질 수 있도록 러닝 커브를 최소화하는 설계를 고민합니다. 모든 팀원이 편하게 작업할 수 있는 코드를 만드는 것을 목표로 삼고 있으며, 저만의 편의보다 팀 전체의 효율을 우선시합니다. 팀 동료의 피드백을 적극 수용하며, 항상 배우는 자세로 개발에 임하고 있습니다. 저는 세상을 더 재미있고 편리하게 만드는 서비스를 구축하는 데 열정을 가지고 있습니다.

## Skills

- C#, C/C++, Unity, TypeScript, Python, Json, Rest api, Dart, Flutter, Objective-c
- Git, Github action, Slack, Notion, ClickUp, Jira
- JetBrains Rider, Visual Studio, Visual Studio Code, Android Studio, Xcode

## Experience(11 년)

2020.11 - 2024.11 (4 년 1 개월)	직방	직방 3d 단지 / Soma(Mac, Win, AOS, IOS)	Client programmer
2018.01 - 2020.10 (2 년 10 개월)	메이커스게임즈	ProjectR(AOS, IOS)	Client programmer
2015.10 - 2017.12 (2 년 4 개월)	위메이드엑스알	이카루스온라인(PC)	Client programmer
2009.09 - 2011.05 (1 년 9 개월)	이야소프트	아이리스온라인(PC)	Client programmer

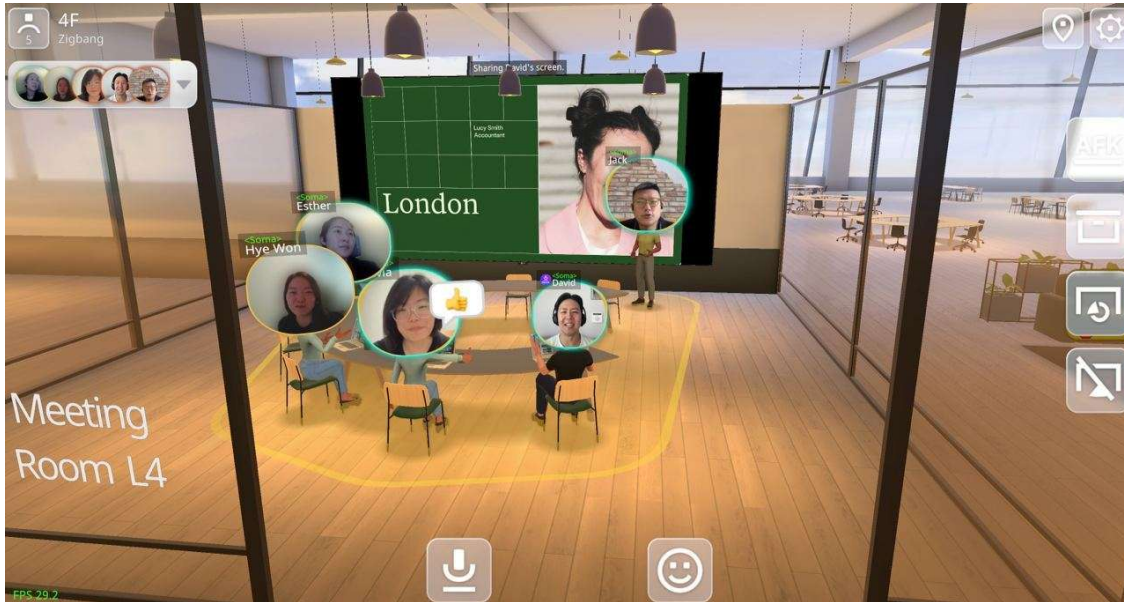
# Work Projects

## Zigbang- 소프트웨어 엔지니어

2020.11 - 2024.11 (4 년 1 개월)

### WebRtc 기반의 3D 아바타 가상 오피스

영상 : <https://www.youtube.com/embed/7iLVSo0nUUU?si=4zMUR9jXgQ1OC8h6>



프로젝트명

Soma

프로젝트 기간

2020.04~2024.11

개발환경

Unity, C#, Typescript, Electron, Python, JetBrains Rider, Xcode, Slack, ClickUp, Github

플랫폼

Windows, Mac, Android, iOS

프로젝트 개요

Unity Engine 기반의 메타버스형 3D 가상 오피스 서비스를 기획 단계부터 출시까지 개발했습니다. Windows, macOS, Android, iOS 멀티 플랫폼을 지원하는 화상 통화 기반 재택근무 솔루션으로, 다수의 기업 고객을 대상으로 상용화했습니다. 초기부터 런칭까지 작업하며 게임적 요소와 업무용 서비스 가치를 융합한 경험을 쌓았습니다.

### 주요 기능 및 담당 업무

#### 1. 커뮤니케이션 시스템 구축

- Agora SDK 기반 WebRTC 화상 솔루션 통합 개발
- 물리 디바이스(웹캠, 마이크, 스피커) 제어 시스템 구현
- 실시간 음성 채팅 및 Speak-to-Text 기능 연동

#### 2. 아키텍처 설계 및 프레임워크 개발

- MV(R)P 아키텍처 기반 UI 프레임워크 설계
- Scene 관리 및 Game Framework 구축

- REST API 통신 및 Packet Handler 구조 설계
  - 체계적인 로깅 시스템 설계
  - UniRx 기반의 반응형 프로그래밍 프로젝트 적용 주도
3. 다수의 콘텐츠 및 UI/UX 개발
    - UGUI 기반 재사용 가능한 UI Component Kit 제작
    - DOTween 을 활용한 다이나믹한 UI 애니메이션 구현
    - 채팅, 인벤토리, 집중 모드, 팀 관리, 대시보드, 관리자창 다수의 주요 콘텐츠 시스템 개발
  4. 크로스 플랫폼 지원
    - macOS, Windows 네이티브 플러그인 개발
    - 플랫폼별 native OS 알림 시스템 구현 (macOS Bundle, Windows DLL)
    - 디바이스 권한 관리 시스템 구축
  5. 기타 기능 구현
    - Barracuda 모델(ONNX) 활용한 실시간 얼굴 인식
    - 다양한 화면 공유 기능 개발 (화면, 카메라, 이미지, 유튜브, 화이트보드)
    - 다국어 지원 시스템
    - 하드웨어 벤치마크 기반 자동 그래픽 품질 최적화

## 주요 기술 적용 경험

### 1. 화면 공유 시스템 구현

WebRTC 기반의 다양한 화면 공유 기능을 구현했습니다. Agora Sdk 를 적용해서 사용자의 전체 화면, 특정 애플리케이션 창, 웹캠 화면등을 공유할 수 있었으며, 이미지, 유튜브 영상, 화이트보드 등 다양한 유형의 콘텐츠를 공유할 수 있는 시스템을 개발했습니다. 상호작용 기능을 활용하여 3D 오브젝트 상에 공유하는 기능도 구현하였습니다.

### 2. 얼굴 인식 및 아바타 연동

Barracuda 모델(ONNX)을 활용해 실시간 얼굴 인식 기능을 구현했습니다. 사용자 얼굴을 실시간으로 인식하여 사용자 자리비움 여부를 감지하여 비대면 업무 환경에서도 대면 환경과 같이 자리비움 상태를 구현하였습니다.

### 3. Native 플러그인 작업

물리 디바이스(웹캠, 마이크, 화면공유)들에 대한 사용 접근 권한으로 수행이 필요하여 macOS 는 Objective-C 언어를 활용해 Bundle 작업을 진행했습니다. 그 이외에 Native 기능을 위해 windows dll 작업등을 진행하여 크로스플랫폼 대응을 하였습니다.

## 프로젝트 회고 및 배운 점

Soma 프로젝트는 기획 단계부터 런칭까지 참여한 의미 있는 경험이었습니다. 코로나 19 팬데믹으로 인한 원격 근무 환경에서 시작된 이 프로젝트는 3D 아바타 기반의 영상 통화를 융합하여 새로운 가상 업무 공간을 제공하는 것이었습니다.

개발 초기에는 게임 엔진을 활용한 업무용 서비스 구현이라는 독특한 접근 방식에 대한 의구심이 있었습니다. 그러나 회사 내부 직원들을 첫 번째 사용자로 삼아 실제 업무 환경에서 지속적으로

테스트하고 피드백을 수집함으로써, 게임적 요소(아바타 조작, 인터랙션)와 업무 기능(화면 공유, 화상 회의)을 효과적으로 융합할 수 있었습니다.

특히 Windows, macOS, Android, iOS 등 다양한 플랫폼에서 일관된 사용자 경험을 "사용 편의성과 실시간성을 높였습니다. 각 OS 별 네이티브 기능(권한 관리, 알림 시스템, 하드웨어 접근)을 통합하는 과정에서 플랫폼 특화 지식을 크게 확장할 수 있었습니다.

거기다 런처 유지보수를 진행하면서 Electron, Typescript 개발환경에 대한 시야를 넓힐 수 있었습니다.

## 1. 커뮤니케이션 시스템



## 2. 화면공유



### 3. 로비 아바타 선택



### 직방 3D 단지 투어 - 실감형 부동산 가상 투어

영상 : [https://www.youtube.com/embed/Wtx45Vvjse4?si=nk1xs0PIVl2y\\_A99](https://www.youtube.com/embed/Wtx45Vvjse4?si=nk1xs0PIVl2y_A99)



프로젝트 기간  
프로젝트 소개  
개발환경  
플랫폼  
프로젝트 개요

2020.11 ~ 2021.04  
Data Driven 방식의 아파트 단지 시뮬레이션  
Unity, C#, Visual Studio Code, Slack, Jira, Github  
Android, iOS

직방의 부동산 중개 플랫폼에 3D 가상 투어 기능을 개발하여 사용자가 직접 방문하지 않고도 현장감 있게



아파트 단지과 호실을 둘러볼 수 있는 서비스를 구현했습니다. Unity 기반의 경량화된 3D 렌더링과 사용자 친화적인 인터페이스를 통해 모바일 환경에서도 빠르고 안정적인 가상 투어 경험을 제공했습니다.

## 주요 기능 및 담당 업무

1. UI/UX 개발
  - UGUI 기반 전체 UI 시스템 개발 및 연출 작업
  - Room 면적별 정보 표시 UX 설계
  - 매물 정보 시각화 시스템 구현
  - 사용자 온보딩을 위한 튜토리얼 시스템 개발
2. 성능 최적화
  - Object Material Pooling 으로 생성 시 지연 최소화
  - 주변 건물 Culling 처리
3. 데이터 통합
  - 서버 API 를 통한 실시간 매물 정보 연동
  - 매물 정보 필터링 및 표시 기능 개발
4. 사용자 경험 개선
  - 다양한 화면 크기 및 해상도 대응

## 주요 기술 적용 경험

1. 모바일 환경 최적화

다양한 기기에서 일관된 성능을 보장하기 위해 Object Pooling 과 Culling 시스템을 세밀하게 조정했습니다. 특히 Object Material Pooling 을 활용하여 생성 시 발생하는 지연을 최소화하고, 화면에 보이지 않는 주변 건물에 대한 효율적인 Culling 처리를 구현함으로써 저사양 모바일 기기에서도 원활한 성능을 확보했습니다.
2. 폴더블 기기 대응

폴더블 기기의 동적 해상도 변경 시 발생하는 렌더링 이슈를 해결했습니다. 화면 접힘 상태가 변경될 때 렌더텍스처가 올바르게 조정되지 않는 문제를 위해, 해상도 변경 이벤트 감지 시 렌더텍스처를 동적으로 재생성하는 방식을 구현했습니다. 다양한 안드로이드 및 iOS 기기에서의 철저한 테스트를 통해 기기별 특성을 고려한 최적화를 진행했습니다.

## 프로젝트 회고 및 배운 점

직방 3D 단지 투어 프로젝트는 게임 엔진 기술을 실용적인 비즈니스 도메인에 적용하는 값진 경험이었습니다. 짧은 개발 기간 내에 높은 완성도를 달성하기 위해 팀원들과 긴밀한 협업 체계를 구축한 것이 성공적인 런칭의 핵심이었습니다.

호, 전경 상세뷰 및 매물정보 시각화

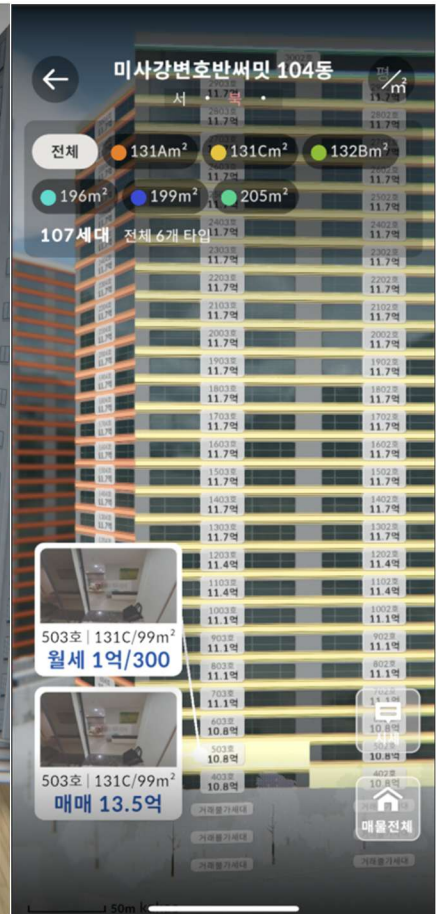
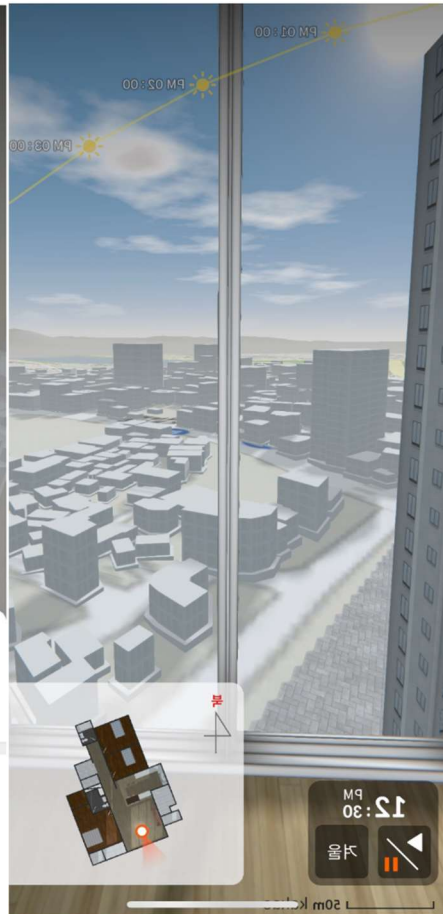


#### 거래불가

이 세대는 지금 매물로 나와 있지 않아요.

1603호 분양가 ? 평당가  
**5억 4,400** **1,605만**/3.3m²

1603호와 같은 타입의 평균 분양가는 5억 2,771입니다.



## 이카루스 온라인 - PC 3D MMORPG

영상 : [https://www.youtube.com/embed/1mp1ZvCoS40?si=gyy\\_ZvbtjOkVKG8S](https://www.youtube.com/embed/1mp1ZvCoS40?si=gyy_ZvbtjOkVKG8S)



프로젝트 기간

2015.10~2017.12

프로젝트 소개

PC 3D Online MMORPG

개발환경

CryEngine3, C++, WinApi, STL, Visual Studio, ActionScript3.0, ScaleForm4.0 Mantis, SVN

플랫폼

Windows

프로젝트 개요

국내에서 성공적으로 서비스 중이던 이카루스 온라인의 북미 런칭 프로젝트에 참여하여 Nexon America 플랫폼 연동 및 북미 시장에 특화된 기능 개발을 담당했습니다. 북미 유저들의 선호도와 사용 패턴을 고려한 다양한 시스템을 구현하여 성공적인 해외 진출에 기여했습니다.

## 주요 기능 및 담당 업무

### 1. Nexon America 플랫폼 연동

- MultiThread Custom Patcher CLI 개발 및 Nexon Launcher 연동
- 안정적인 패치 시스템 구현으로 런칭 초기 이탈률 감소
- Nexon America 로그인 시스템 연동

### 2. 북미 특화 UI/UX 개선

- 액션 모드 시스템 개발 (크로스헤어 모드, 퀵슬롯 추가, 타겟 유지 시스템)
- UI 크기 조절 옵션 및 저사양 모드 설정 구현



- 퀘스트 네비게이션 시스템 개발 (목표 지점 방향 및 거리 표시)
- 북미 징계 처리 표시 시스템 구현 (계정 제재 만료 기간 및 사유 표시)

### 3. 게임 콘텐츠 개발 및 최적화

- 3:3 PvP 전장 시스템, 대규모 유물 쟁탈전 PvP 콘텐츠, 통합 던전 용병단 PvE 콘텐츠 구현
- 캐시샵 시스템 개선 (캐시 타입 추가, 아이템 레벨 제한)
- 아이템 강화 시스템 개선
- 펫 길들이기/컬렉션 개선, 길드 가입/홍보
- 대규모 UI/UX 개편 작업

### 4. 시스템 안정화

- 네트워크 안정성 향상 (디스커넥 리커넥팅)
- CryPack 분할 패치 개선

### 5. 로컬라이징 시스템 개선

- Excel Automation Library 기반 텍스트 관리 시스템 개발
- 국가별 이미지 자동 교체 시스템 구현

## 주요 기술 적용 경험

### 1. CryPack 파일 분할 패치 시스템

CryEngine 의 Pak 파일 제한용량(1.9GB) 초과 시 패치가 되지 않는 심각한 이슈가 있었습니다. 이를 해결하기 위해 제한용량 초과 시 신규 Pak 파일을 생성하고 자동으로 패치되는 시스템을 구현했습니다. 특히 다중 스레드 기반의 Custom Patcher 를 개발하여 Nexon Launcher 와 성공적으로 연동함으로써 패치 안정성과 속도를 크게 개선했습니다.

### 2. 네트워크 리커넥팅 시스템

네트워크 단절 시 기존에는 로비를 재진입해야 했던 불편함을 개선하기 위해, 서버 간 직접 리커넥팅 시스템을 구현했습니다. 이를 통해 로딩 시간을 단축하고 사용자 경험을 크게 향상시켰습니다.

## 프로젝트 회고 및 배운 점

이카루스 온라인의 북미 런칭 TF 팀에 합류하여 글로벌 서비스 준비 과정에서 중요한 역할을 수행했습니다. 본 프로젝트에서 가장 가치 있었던 경험은 Nexon America 의 플랫폼 연동을 통한 북미 서비스 런칭이었습니다. 특히, 다중 스레드 기반의 Custom Patcher 를 개발하는 과정에서 네트워크 프로그래밍과 멀티스레딩에 대한 심도 있는 지식을 습득할 수 있었으며, 이는 이후 프로젝트에서도 큰 자산이 되었습니다. Nexon America 와의 지속적인 협업 과정에서 글로벌 서비스 개발에 대한 깊은 이해를 얻게 되었으며, 결과적으로 기존 국내 서비스보다 향상된 시스템을 구축하여 성공적인 북미 런칭을 달성했습니다. 이 경험은 글로벌 시장을 타겟으로 하는 게임 개발의 중요한 이정표가 되었습니다.

## 거점 콘텐츠

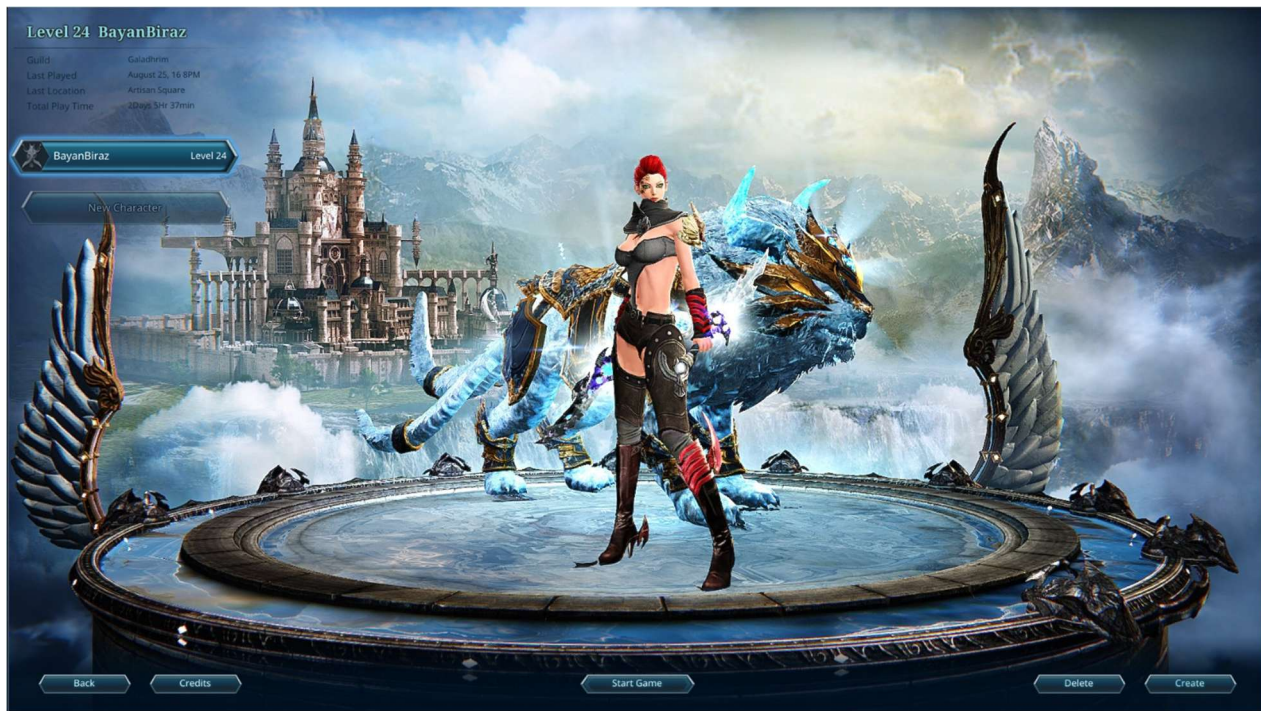


## 펠로우 소환수





## 로비 캐릭터 선택



아이리스 온라인 - PC 3D MMORPG

영상 : <https://www.youtube.com/embed/koKhtPk20s4?si=mcemeQ2VktE1Kf4b>



프로젝트 기간	2009.09~2011.05
프로젝트 소개	PC 3D Online MMORPG
개발 환경	C++, WinApi, STL, Visual Studio, MFC, GameBryo, Perforce
플랫폼	Windows
프로젝트 개요	

아이리스 온라인의 서비스 런칭 3 개월 전 합류하여 상용화 준비 및 러시아 진출을 위한 현지화 작업을 담당했습니다. 초기에는 클라이언트 개발에 집중하다가 점차 서버 및 데이터베이스 작업까지 범위를 확장하여 풀스택 개발 경험을 쌓았습니다.

주요 기능 및 담당 업무

1. SNS 연동 시스템 개발
  - Twitter Open API 및 OAuth 기반 인증 시스템 구현
  - 게임 내에서 트윗 작성 및 피드 리스트 실시간 표시 기능 개발
  - 게임 이벤트와 연동된 자동 SNS 알림 시스템 구축
2. 핵심 콘텐츠 개발
  - 파티/길드 시스템 개선
  - 친구 시스템 구현
  - 캐시 아이템 개발 (외형 변경, 탈것)
  - 콘텐츠 단위의 서버, 클라이언트, DB(MSSQL) Stored Procedure 통합 개발
3. 글로벌 서비스 대응



- 러시아 서비스 현지화 및 런칭 지원
- 패치 작업 및 유지 보수
- 국가별 정책에 맞는 게임 시스템 조정

## 주요 기술 적용 경험

### 1. Twitter API 연동

당시 소셜 네트워크의 게임 내 통합은 일반적이지 않았습니다. OAuth 인증 방식과 REST API 를 활용하여 게임 내에서 Twitter 기능을 구현할 수 있었습니다. 특히 게임 클라이언트와 웹 서비스 간의 통신을 안정적으로 구현하기 위해 Thread 를 활용하여 비동기 통신 방식을 적용하고, 오류 처리 및 재시도 메커니즘을 구축했습니다.

### 2. 풀스택 개발 경험

초기 클라이언트 개발에서 시작하여 점차 서버와 데이터베이스(MSSQL Stored Procedure) 작업까지 담당하게 되면서, 콘텐츠 시스템 단위에 아키텍처를 이해하고 개발할 수 있는 역량을 키웠습니다.

## 프로젝트 회고 및 배운 점

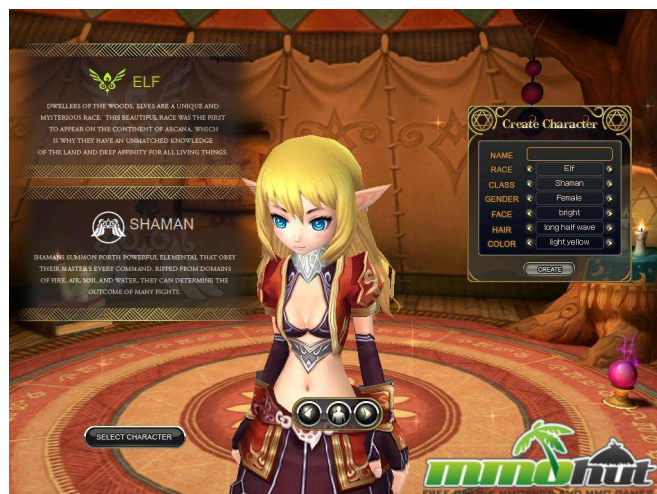
아이리스 온라인 프로젝트는 상용 서비스 직전 단계와 해외 진출 과정에 참여한 귀중한 경험이었습니다. 특히 Twitter API 연동 작업이 가장 흥미롭고 의미 있는 업무였습니다. OAuth 인증 방식과 REST API 활용을 통해 게임 내에서 소셜 기능을 구현하는 과정에서 웹 서비스 연동에 대한 실질적인 경험을 얻을 수 있었습니다. 또한 초기 클라이언트 개발에서 시작하여 점차 서버와 데이터베이스 작업까지 담당하게 되면서, MMORPG 의 전체 시스템 아키텍처를 이해하고 능력을 기를 수 있었습니다. 글로벌 서비스를 위한 현지화 과정에서는 다국어 지원을 고려한 개발의 중요성을 체득했습니다.

이 프로젝트를 통해 게임 서비스의 전 주기에 걸친 이해와 함께, 다양한 기술 스택을 활용한 풀스택 개발 역량을 쌓을 수 있었습니다. 특히 런칭 준비 단계에서의 집중적인 디버깅과 최적화 작업은 이후 프로젝트에서도 큰 도움이 되었습니다.

## 1. 콘텐츠



## 2. 트위터 연동 및 로비 캐릭터 화면



# Personal Projects

## 뿌요뿌요 퍼즐 게임



**개발 환경:** C++, SDL3, Visual Studio 2022

**플랫폼:** Windows

**GitHub:** [https://github.com/aquhm/puzzle\\_puyopuyo](https://github.com/aquhm/puzzle_puyopuyo)

### 프로젝트개요

Modern C++ 기능과 SDL3 라이브러리를 활용하여 네트워크 대전이 가능한 뿌요뿌요 스타일의 퍼즐 게임을 개발했습니다. p2p 기반의 서버 및 클라이언트를 모두 구현하여 1:1 대전이 가능하게 개발하였습니다.

### 주요 기능 및 담당 업무

#### 1. P2P 네트워크 대전 시스템

- 한 사용자가 서버 역할을 하고 다른 사용자가 클라이언트가 되어 실시간으로 대전할 수 있는 구조 설계
- 패킷 기반 통신과 비동기 I/O 를 활용한 효율적인 네트워크 처리
- 각 패킷 타입별 전용 프로세서로 모듈화된 패킷 처리 시스템 구현

#### 2. 게임 시스템 설계

- 상태 패턴을 활용한 게임의 다양한 화면과 상태 전환 관리
- 리소스, 상태, 플레이어 등을 관리하는 매니저 클래스 설계
- 시각적 효과를 위한 파티클 시스템과 애니메이션 구현



## 주요 기술 적용 경험

### 1. 시스템 아키텍처 설계

다양한 디자인 패턴을 적용하여 확장성 있고 모듈화된 시스템 아키텍처를 구현했습니다. 상태 패턴을 통해 게임의 다양한 화면과 상태 전환을 관리하고, 매니저 시스템으로 리소스와 게임 객체를 효율적으로 관리했습니다. 특히 네트워크 통신과 게임 로직을 분리하여 각 시스템의 독립성을 보장하면서도 유기적인 상호작용이 가능하도록 설계했습니다.

### 재귀 함수를 활용한 매칭 체크

게임 로직상 블록 체크를 효과적으로 해야하는데, 이 과정에서 재귀 로직을 활용하여 Block 체크 및 삭제해야할 방해블록을 선별 추출하였습니다.

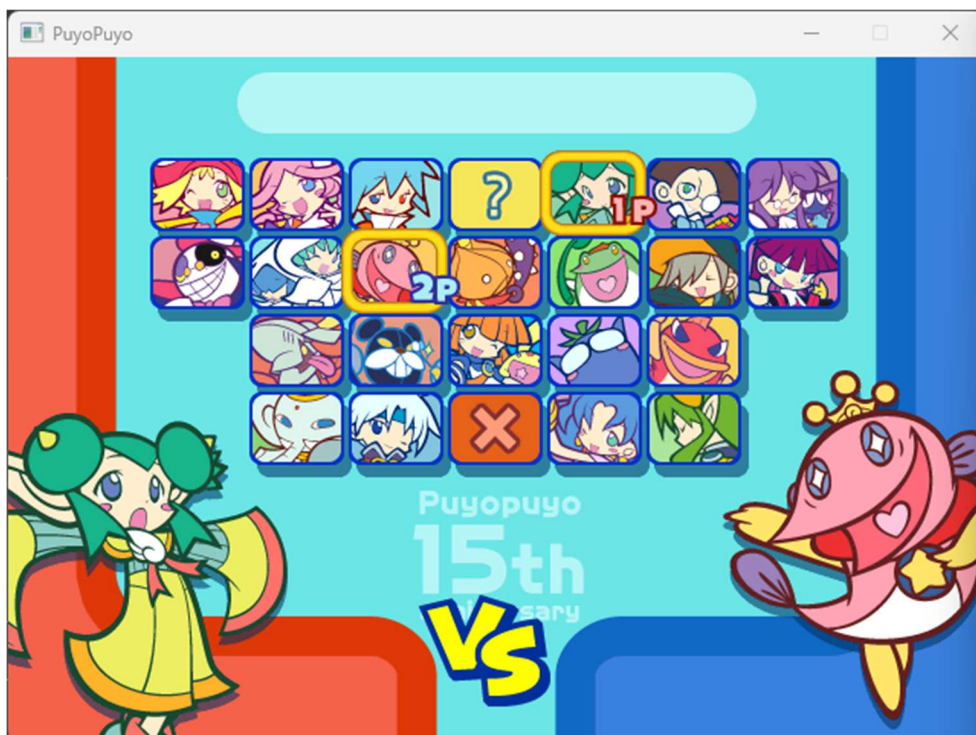
### 2. 네트워크 프로그래밍

P2P 기반의 네트워크 대전 기능을 구현하면서 Windows Socket API 와 IOCP(I/O Completion Port)를 활용한 비동기 네트워크 프로그래밍에 대한 이해도를 높였습니다.

## 프로젝트 회고 및 배운 점

이 프로젝트를 통해 SDL3 엔진 프레임워크에 좀 더 익숙해지고 숙지할 수 있었습니다. 학생 시절 친구들과 즐겨하던 게임을 직접 만들고 기능을 단계적으로 구현하면서 보람과 재미를 느꼈습니다. 클라, 서버 풀스택으로 진행하면서 패킷을 정의하고 동기화는 과정 또한 즐겁고 보람있었습니다.

### 1. 캐릭터 선택 화면

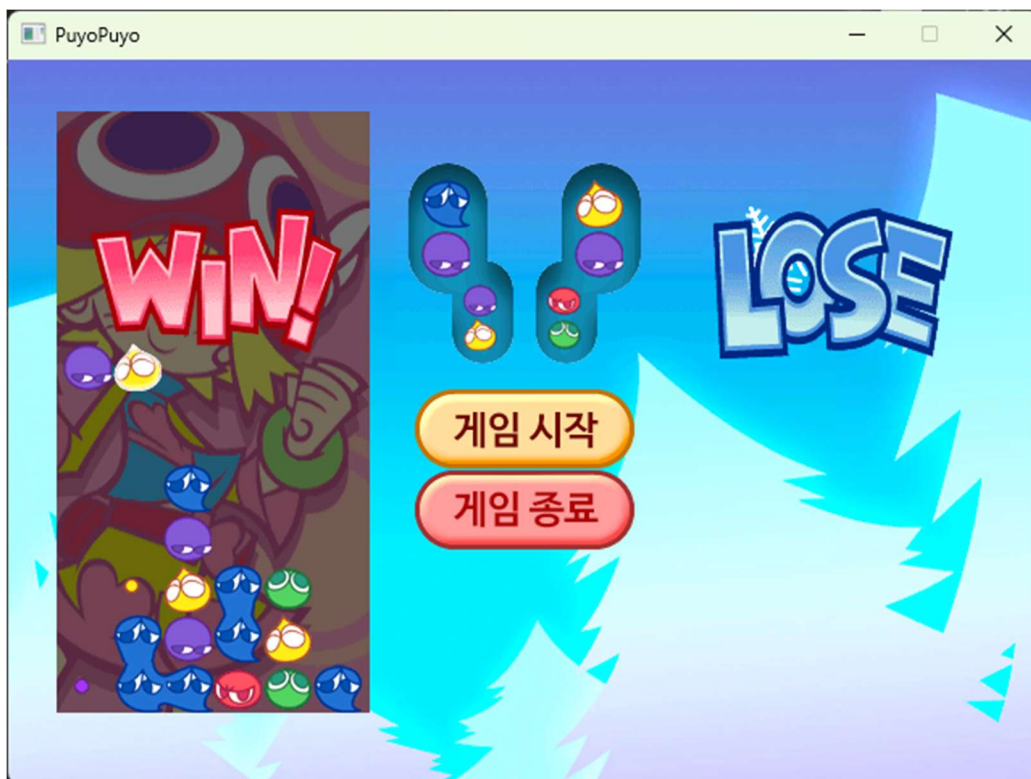




## 2. 게임 화면

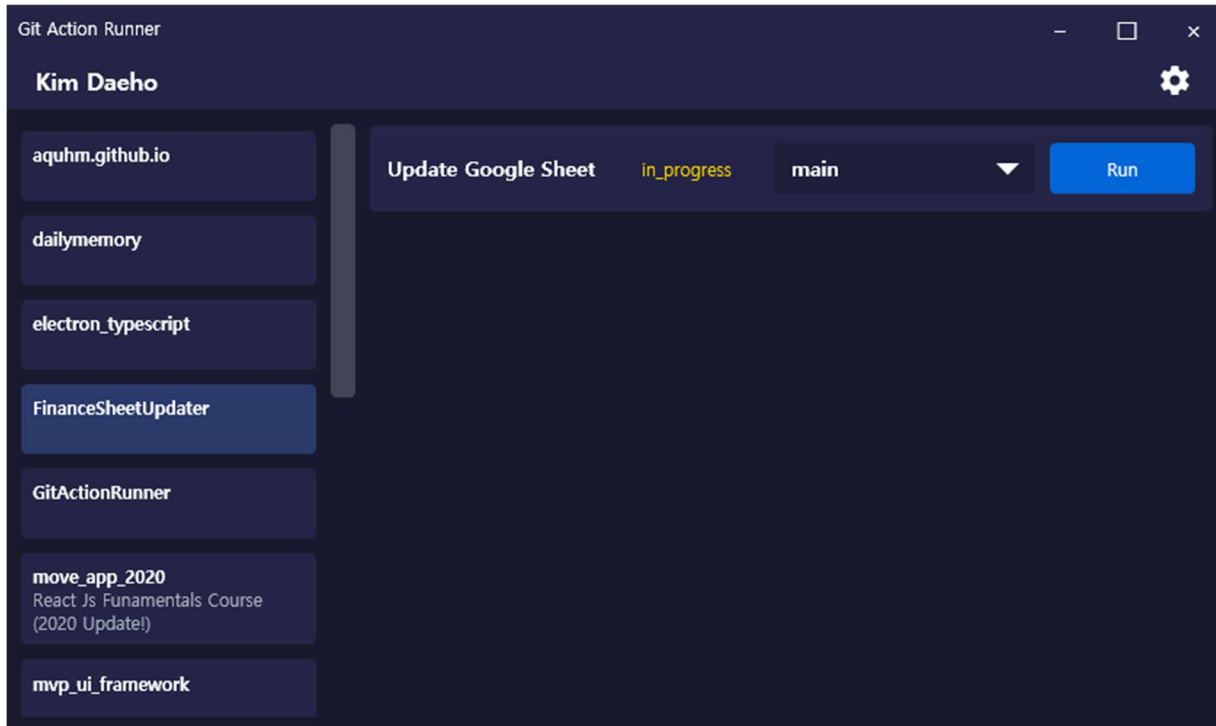


## 3. 결과 화면



영상 : <https://www.youtube.com/embed/c-e-KC9jHfY>

# Git Action Runner



**개발 환경:** C#, .NET 9.0, WPF, Visual Studio 2022

**플랫폼:** Windows

**GitHub:** <https://github.com/aquhm/GitActionRunner>

## 프로젝트개요

Git Action Runner 는 Windows 환경에서 GitHub Actions 워크플로우를 관리할 수 있는 데스크톱 애플리케이션입니다. 개발자들이 복잡한 CI/CD 파이프라인을 쉽게 모니터링하고 제어할 수 있도록 직관적인 인터페이스를 제공합니다. Personal Access Token 기반의 안전한 인증 시스템, 실시간 워크플로우 상태 모니터링, 브랜치 관리 등 다양한 기능을 통해 개발 생산성을 향상시킵니다.

## 주요 기능 및 담당 업무

1. GitHub API 연동
  - Personal Access Token 기반 원클릭 인증 구현
2. 워크플로우 관리 시스템
  - 실시간 워크플로우 실행 상태 모니터링 기능 구현
  - 리포지토리별 브랜치 선택 및 워크플로우 실행 기능 개발
3. 사용자 인터페이스 최적화
  - Windows 네이티브 알림 시스템 통합
  - 워크플로우 상태에 따른 시각적 피드백 제공

## 주요 기술 적용 경험

### 1. MVVM 아키텍처 구현

WPF MVVM 패턴 구조로 ViewModels 에서 Commands 와 Properties 바인딩을 통해 View 와의 느슨한 결합을 유지하는데, 집중했습니다. 또한 Microsoft.Extensions.DependencyInjection 을 활용한 의존성 주입 시스템을 구축하여 컴포넌트 간의 결합도를 낮추고 모듈화된 구조를 만들었습니다.

### 2. GitHub API 연동

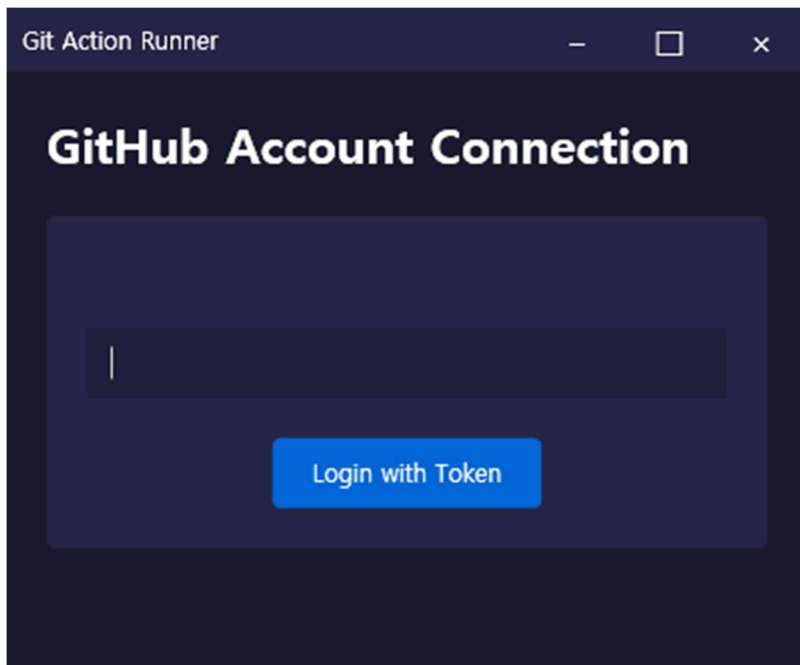
Octokit.NET 라이브러리를 사용하여 GitHub API 와 통합하는 과정에서 REST API 호출, 비동기 프로그래밍, 예외 처리 등의 스킬을 향상시켰습니다.

## 프로젝트 회고 및 배운 점

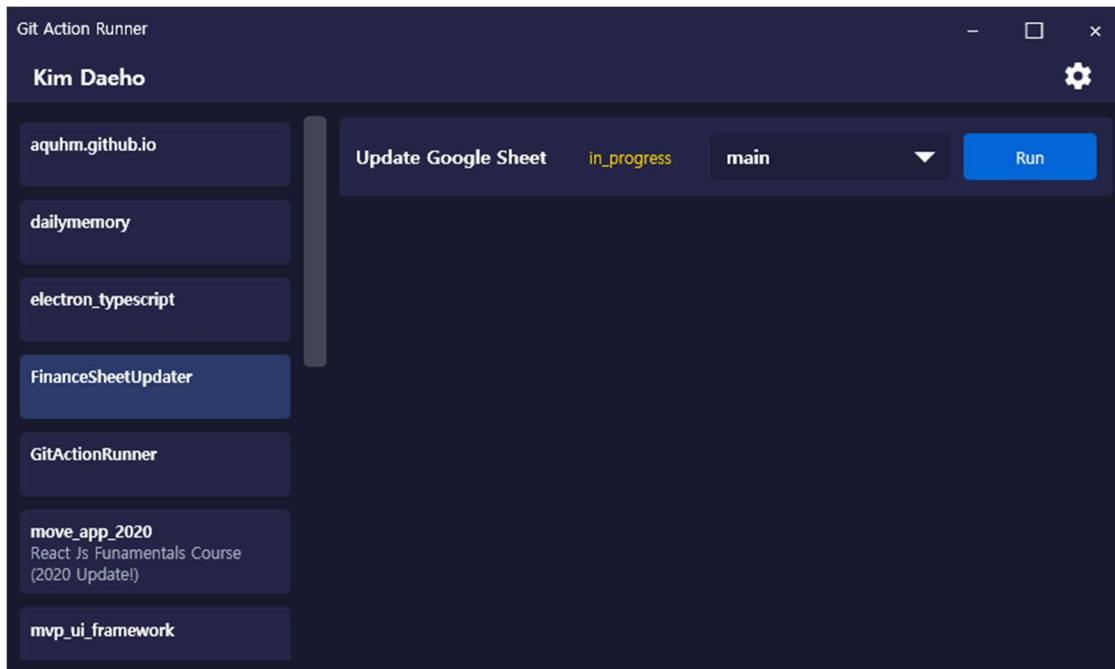
프로젝트를 진행하면서 CI/CD 파이프라인 상태를 확인하기 위해 GitHub 웹 인터페이스를 여러 단계에 걸쳐 탐색해야 하는 번거로움을 경험했습니다. 이러한 불편함을 해소하고자 데스크톱 환경에서 직접 워크플로우를 모니터링할 수 있는 애플리케이션의 필요성을 인식하게 되었습니다. 특히 CI/CD 워크플로우가 완료되었을 때 Windows 네이티브 알림을 통해 즉시 결과를 확인할 수 있다면 개발 생산성이 크게 향상될 것으로 판단했습니다.

GitHub API 를 활용하여 워크플로우 상태 정보를 조회하고, WPF 프레임워크를 통해 이 정보를 실시간으로 시각화할 수 있었습니다.

### 1. 로그인 화면



## 2. 워크플로우 상태 화면





# SheetGenerator

@table:ClientTest						
테스트 테이블입니다. (개발용)						
Index	Key	Description	Column1	Column2	Column3	Column4
인덱스(필수)	스트링 키	설명	설명1	설명2	설명3	설명4
int ▼	string ▼	float ▼	int ▼	string ▼	bool ▼	double ▼
1	key_1	1	1	a	true	123
2	key_2	2	2	b	false	123
3	key_3	3	3	c	true	123
4	key_4	4	4	a	false	123
5	key_5	5	5	b	true	123
6	key_6	6	6	c	false	123
7	key_7	7	7	a	true	123
8	key_8	8	8	b	false	123
9	key_9	9	9	c	true	123
10	key_10	10	10	a	false	123
11	key_11	11	11	b	true	123

개발 환경: C# 10.0, .NET 7.0, Visual Studio 2022

플랫폼: Windows

GitHub: <https://github.com/aquhm/SheetGenerator>

## 프로젝트 개요

SheetGenerator 는 구글 시트나 Excel 데이터를 바이너리 또는 JSON 형식으로 변환하고, 해당 데이터를 효율적으로 관리할 수 있는 C# 코드를 자동으로 생성하는 도구입니다. 게임 개발 및 응용 프로그램에서 자주 필요한 데이터 관리 솔루션으로, 기획자와 개발자 간의 협업을 원활하게 만들어줍니다.

## 주요 기능 및 담당 업무

- 데이터 импорт 및 변환 시스템
  - 구글 API 를 통해 스프레드시트 데이터를 직접 가져오는 기능 구현
  - 다양한 형식(바이너리, JSON)으로 데이터 변환 모듈 개발
- 코드 생성 엔진
  - 템플릿 기반 코드 생성 시스템 설계 및 구현
  - 데이터 접근을 위한 클래스와 메서드 자동 생성
  - 확장 가능한 템플릿 시스템 개발
- 데이터 최적화 및 성능 향상
  - MessagePack 을 이용한 효율적인 바이너리 직렬화/역직렬화 구현
  - Index 와 Key 기반 데이터 검색 알고리즘 최적화

## 주요 기술 적용 경험

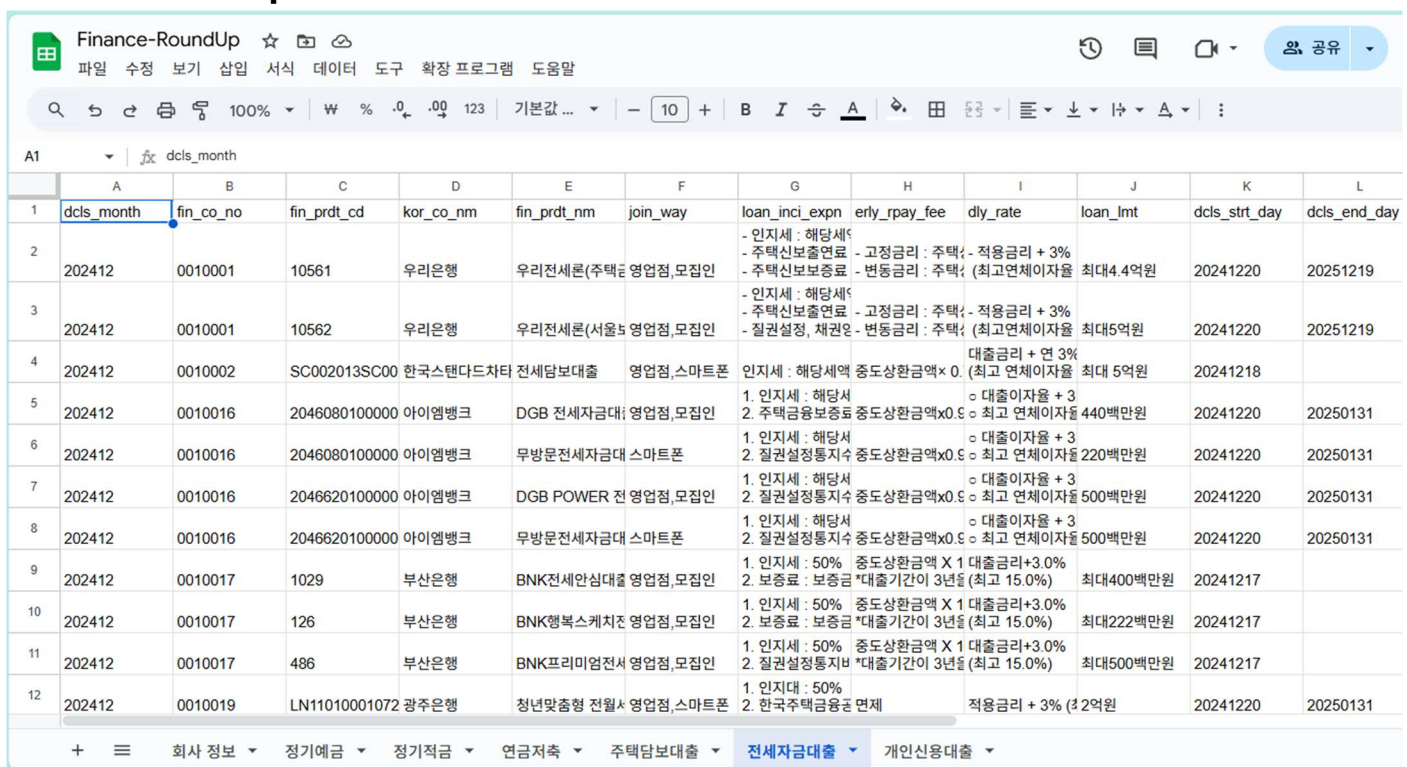
- 확장성 있는 아키텍처 설계

다양한 데이터 소스와 출력 형식, 그리고 여러 프로그래밍 언어를 지원하기 위한 확장 가능한 아키텍처를 설계하는 것이 목표이었습니다. 인터페이스와 팩토리 패턴을 활용하여 새로운 기능을 추가할 때 기존 코드를 수정하지 않고도 확장할 수 있는 구조를 만들었습니다. C#뿐만 아니라 C++, Python 등 다른 언어 코드로 생성가능하게 확장성 있게 추상화하여 구조화하였습니다.

## 프로젝트 회고 및 배운 점

실제 프로젝트를 진행하면서 기획자와 개발자 간의 협업 효율성과 협업성의 중요성을 알고 있었습니다. 반복적인 작업을 자동화하면 프로젝트 효율성이 좋아지기 때문에 그런 경험을 토대로 만들어 보았습니다. 현재는 구글시트만 지원하지만, 추후 Excel 파일로부터 진원하도록 기능 추가 예정입니다.

## FinanceSheetUpdater



	A	B	C	D	E	F	G	H	I	J	K	L
1	dcls_month	fin_co_no	fin_prdt_cd	kor_co_nm	fin_prdt_nm	join_way	loan_inci_expn	erly_rpay_fee	dly_rate	loan_lmt	dcls_strt_day	dcls_end_day
2	202412	0010001	10561	우리은행	우리전세론(주택)	영업점,모집인	- 인지세 : 해당세 - 주택신보출연료 - 주택신보보증료	- 고정금리 : 주택 - 변동금리 : 주택	- 적용금리 + 3% (최고연체이자율)	최대4.4억원	20241220	20251219
3	202412	0010001	10562	우리은행	우리전세론(서울)	영업점,모집인	- 인지세 : 해당세 - 주택신보출연료 - 주택신보보증료	- 고정금리 : 주택 - 변동금리 : 주택	- 적용금리 + 3% (최고연체이자율)	최대5억원	20241220	20251219
4	202412	0010002	SC002013SC00	한국스탠다드차타	전세담보대출	영업점,스마트폰	인지세 : 해당세액	중도상환금액x0.0	대출금리 + 연 3% (최고 연체이자율)	최대 5억원	20241218	
5	202412	0010016	2046080100000	아이엠뱅크	DGB 전세자금대	영업점,모집인	1. 인지세 : 해당세 2. 주택금융보증료	중도상환금액x0.0	대출이자율 + 3% (최고 연체이자율)	440백만원	20241220	20250131
6	202412	0010016	2046080100000	아이엠뱅크	무방문전세자금대	스마트폰	1. 인지세 : 해당세 2. 집권설정통지수	중도상환금액x0.0	대출이자율 + 3% (최고 연체이자율)	220백만원	20241220	20250131
7	202412	0010016	2046620100000	아이엠뱅크	DGB POWER 전	영업점,모집인	1. 인지세 : 해당세 2. 집권설정통지수	중도상환금액x0.0	대출이자율 + 3% (최고 연체이자율)	500백만원	20241220	20250131
8	202412	0010016	2046620100000	아이엠뱅크	무방문전세자금대	스마트폰	1. 인지세 : 해당세 2. 집권설정통지수	중도상환금액x0.0	대출이자율 + 3% (최고 연체이자율)	500백만원	20241220	20250131
9	202412	0010017	1029	부산은행	BNK전세안심대출	영업점,모집인	1. 인지세 : 50% 2. 보증료 : 보증금	중도상환금액 X 1 대출기간이 3년(최고 15.0%)	대출금리+3.0% (최고 연체이자율)	최대400백만원	20241217	
10	202412	0010017	126	부산은행	BNK행복스캐치	영업점,모집인	1. 인지세 : 50% 2. 보증료 : 보증금	중도상환금액 X 1 대출기간이 3년(최고 15.0%)	대출금리+3.0% (최고 연체이자율)	최대222백만원	20241217	
11	202412	0010017	486	부산은행	BNK프리미엄전세	영업점,모집인	1. 인지세 : 50% 2. 집권설정통지비	중도상환금액 X 1 대출기간이 3년(최고 15.0%)	대출금리+3.0% (최고 연체이자율)	최대500백만원	20241217	
12	202412	0010019	LN11010001072	광주은행	청년맞춤형 전월세	영업점,스마트폰	1. 인지대 : 50% 2. 한국주택금융공	연체	적용금리 + 3% (3.2억원)		20241220	20250131

**개발 환경:** Python 3.10, AWS S3, Google Sheets API, Visual Studio Code, Git

**플랫폼:** Windows

**GitHub:** <https://github.com/aquhm/FinanceSheetUpdater>

## 프로젝트 개요

금융감독원의 금융상품 API, 환율 정보, 금속 시세 등 다양한 금융 데이터를 자동으로 수집하여 Google Sheets 와 AWS S3 에 저장하는 파이썬 기반 시스템을 개발했습니다. 데이터 수집, 변환, 저장의 전체 과정을 자동화하여 최신 금융 정보에 쉽게 접근 및 각종 개발환경에서 변환한 Json, csv 를 접근하여 활용할 수 있도록 구축하였습니다.

## 주요 기능 및 담당 업무

### 1. 금융 데이터 수집 모듈 개발

- 구글 API 를 통해 스프레드시트 데이터를 직접 가져오는 기능 구현

- 한국수출입은행 API 를 통한 최신 환율 데이터 자동 수집
- FinanceDataReader 라이브러리를 활용한 금속 및 원자재 시세 정보 수집

## 2. Google Sheets 연동 시스템

- Google API 서비스 계정을 활용한 인증 시스템 구현
- gspreload 라이브러리를 활용한 데이터 업데이트 기능 개발

## 3. AWS S3 저장 시스템

- Boto3 라이브러리를 활용한 AWS S3 업로드 모듈 개발
- JSON 및 CSV 형식의 데이터 자동 저장 및 업데이트 기능

## 주요 기술 적용 경험

### 1. API 호출 최적화

금융감독원 API 는 일일 요청 수 제한이 있어, 매번 요청하는 것이 아니라 일정 시간마다 요청한 데이터를 Google Sheets 에 업로드 후 그 data 를 json, csv 로 변환해서 AWS s3 에 업로드 하도록 하였습니다.

## 프로젝트 회고 및 배운 점

Python 생태계의 다양한 라이브러리와 API 를 활용한 데이터 파이프라인 구축 경험을 쌓을 수 있었습니다. AWS 클라우드 서비스, Google Sheets API 연동 및 GitHub Action CI/CD 를 Workflow 설정으로 데이터 수집, 변환, 업로드 등의 과정을 자동화할 수 있었습니다.

## 1. AWS S3 저장 상태 화면

Amazon S3

- 범용 버킷
- 디렉터리 버킷
- 데이터 버킷
- Access Grants
- 엑세스 지점
- 객체 Lambda 엑세스 지점
- 다중 리전 엑세스 지점
- 배치 작업
- S3 IAM Access Analyzer

이 계정의 퍼블릭 엑세스 차단 설정

▼ Storage Lens

- 대시보드
- Storage Lens 그룹
- AWS Organizations 설정

기능 소프트웨어 11

▶ S3용 AWS Marketplace

weight500-finance 정보

객체 속성 권한 지표 관리 엑세스 지점

객체 (17)

객체는 Amazon S3에 저장되어 있는 기본 엔티티입니다. [Amazon S3 인벤토리](#)를 사용하여 버킷에 있는 모든 객체의 목록을 얻을 수 있습니다. 다른 사용자가 객체에 액세스할 수 있게 하려면 명시적으로 권한을 부여해야 합니다. [자세히 알아보기](#)

Q. 첨부자료 객체 찾기

<input type="checkbox"/>	이름	▲	유형	▼	마지막 수정	▼	크기	▼	스토리지 클래스	▼
<input type="checkbox"/>	<a href="#">all_metal_prices.json</a>		json		2025. 5. 11. pm 9:00:08 PM KST		78.3KB		Standard	
<input type="checkbox"/>	<a href="#">annuity.csv</a>		csv		2025. 5. 11. pm 9:19:38 PM KST		1.8MB		Standard	
<input type="checkbox"/>	<a href="#">annuity.json</a>		json		2025. 5. 11. pm 9:19:37 PM KST		6.5MB		Standard	
<input type="checkbox"/>	<a href="#">bank_logo.json</a>		json		2024. 9. 25. pm 8:41:49 PM KST		2.7KB		Standard	
<input type="checkbox"/>	<a href="#">company.csv</a>		csv		2025. 5. 11. pm 9:18:38 PM KST		97.7KB		Standard	
<input type="checkbox"/>	<a href="#">company.json</a>		json		2025. 5. 11. pm 9:18:36 PM KST		325.8KB		Standard	
<input type="checkbox"/>	<a href="#">credit.csv</a>		csv		2025. 5. 11. pm 9:19:51 PM KST		38.0KB		Standard	
<input type="checkbox"/>	<a href="#">credit.json</a>		json		2025. 5. 11. pm 9:19:50 PM KST		141.9KB		Standard	
<input type="checkbox"/>	<a href="#">deposit.csv</a>		csv		2025. 5. 11. pm 9:19:11 PM KST		297.7KB		Standard	
<input type="checkbox"/>	<a href="#">deposit.json</a>		json		2025. 5. 11. pm 9:19:10 PM KST		783.0KB		Standard	
<input type="checkbox"/>	<a href="#">exchange_rates.json</a>		json		2024. 10. 14. pm 3:27:40 PM KST		5.1KB		Standard	
<input type="checkbox"/>	<a href="#">mortgage.csv</a>		csv		2025. 5. 11. pm 9:19:42 PM KST		77.4KB		Standard	
<input type="checkbox"/>	<a href="#">mortgage.json</a>		json		2025. 5. 11. pm 9:19:42 PM KST		170.1KB		Standard	
<input type="checkbox"/>	<a href="#">rent.csv</a>		csv		2025. 5. 11. pm 9:19:46 PM KST		30.7KB		Standard	
<input type="checkbox"/>	<a href="#">rent.json</a>		json		2025. 5. 11. pm 9:19:46 PM KST		65.3KB		Standard	
<input type="checkbox"/>	<a href="#">saving.csv</a>		csv		2025. 5. 11. pm 9:19:20 PM KST		239.4KB		Standard	
<input type="checkbox"/>	<a href="#">saving.json</a>		json		2025. 5. 11. pm 9:19:20 PM KST		586.1KB		Standard	