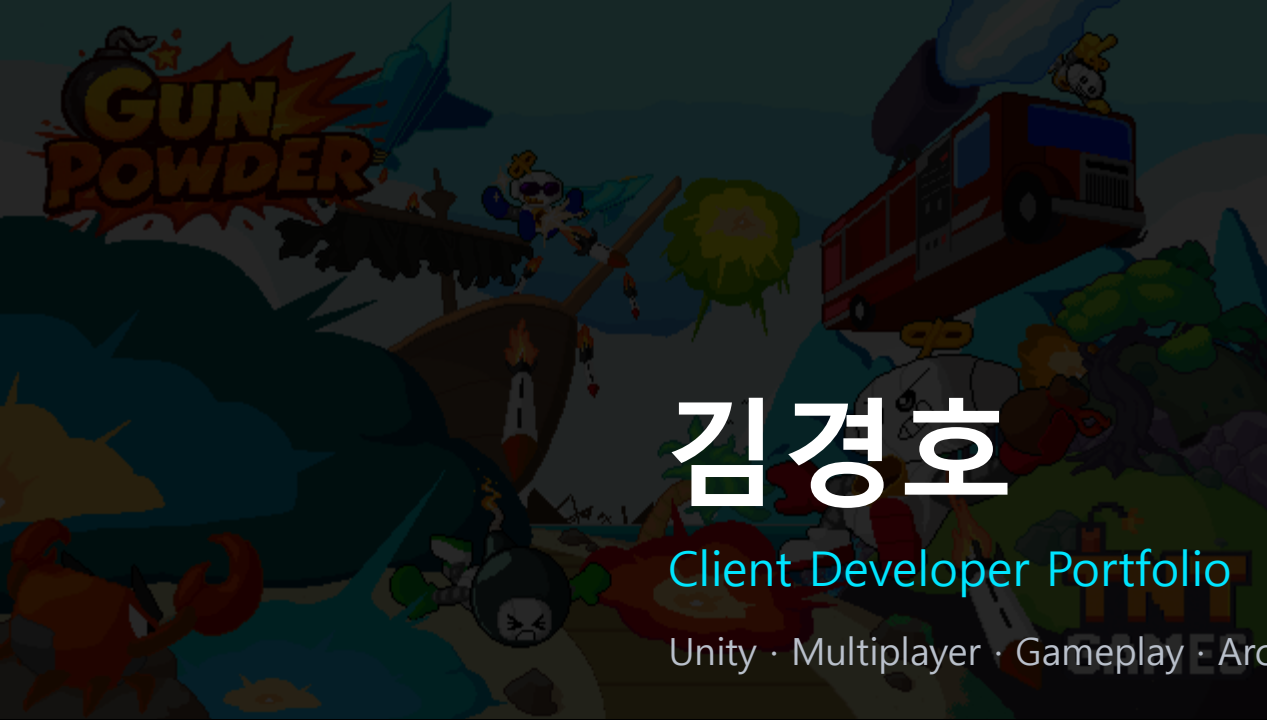




김경호

Client Developer Portfolio

Unity · Multiplayer · Gameplay · Architecture



Profile

안녕하세요, 클라이언트 개발자를 꿈꾸는 김경호입니다.



2015 ~ 2021

수원과학대학교 컴퓨터정보과 졸업

2012 ~ 2015

배문고등학교 졸업

자격증

- 정보처리기능사
- 네트워크관리사 2급

해외연수

- 2018.08 ~ 2019.08
- 일본 워킹홀리데이

-  Email kimkyoungho0726@gmail.com

-  GitHub <https://github.com/guardkim>



Competences

역량의 전문성에 따라 상 · 중 · 하 수준으로 분류하였습니다.

Hard Skills

상

- 인게임 요소 설계, 구현
- 아웃게임 요소 설계, 구현
- 아키텍처 설계, 구현

중

- Photon PUN2

Soft Skills / Tools

상

- 커뮤니케이션
- 협업 능력
- 문서화
- 책임감

중

- Unity6
- Unreal Engine 5

하

- DirectX 11

Microsoft®
DirectX[®]11





교육 사항

2025.02 ~ 2025.08

Com2us SKKU SAY 1기 수료

생성형AI기반 게임콘텐츠 개발 실무 과정
(C#, Unity6 2D, 3D 게임 개발)

2022.12 ~ 2023.06

청년취업사관학교

C++기반 언리얼엔진 게임 개발자 양성 과정
(C++, DirectX2D, DirectX3D, Unreal Engine 5 C++ 3D게임 개발)

2021.07 ~ 2022.01

서울게임아카데미

게임 엔진 프로그래머 양성 과정
(DirectX 3D, Unreal Engine 4.24 C++ 3D게임 개발)

2020.12 ~ 2021.06

서울게임아카데미

게임 프로그래밍 기초
(C, C++, WinAPI, DirectX 2D게임 개발)



경력

2025.10 ~ 2026.04

라이크코퍼레이션

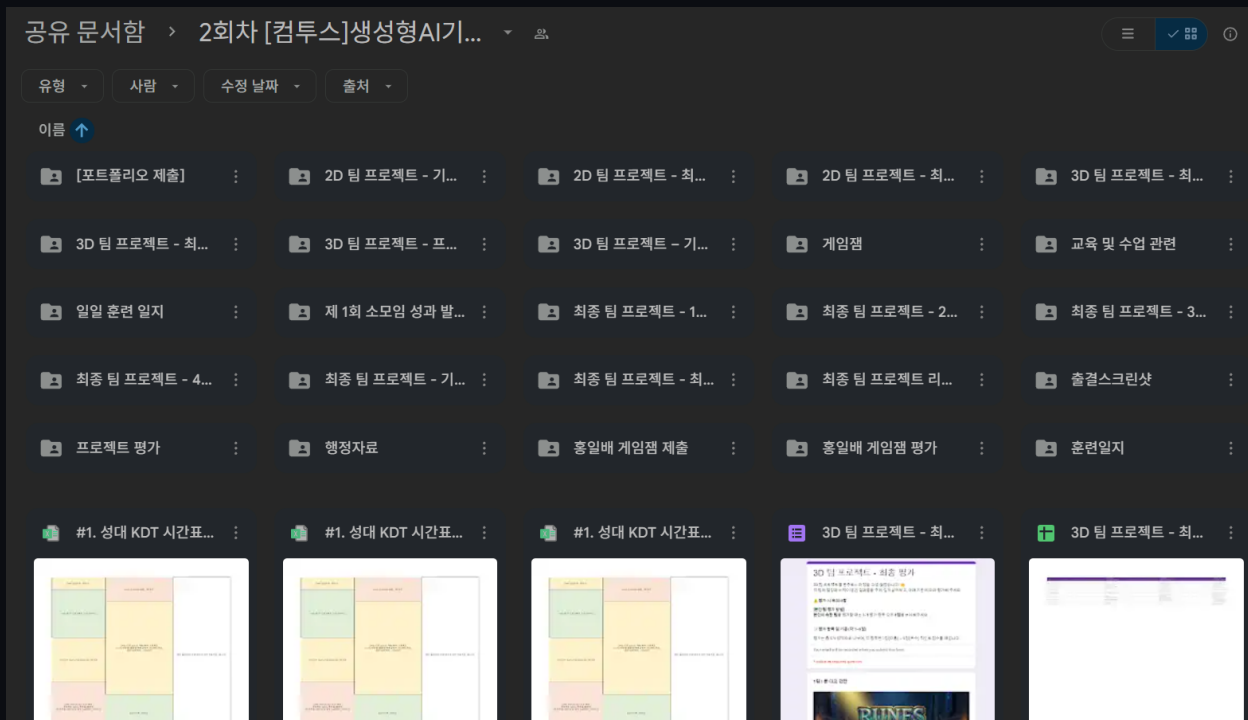
담당업무

- Com2us SKKU SAY 2기

학생 출결 관리, 수업 보조자료 구현, 가이드 작성

SAY 2기 자료

- 2D 프로젝트 계획서
- 최종 프로젝트 선호도 조사
- SAY 프로젝트 게임 플레이 영상 링크 모음
- 3D 팀 프로젝트 - 결과 발표
- Unity 웹 빌드 및 itch.io 배포
- Unity Play 배포
- 클릭잼 피드백
- SKKU 2기 과제 목록
- Gemini 연동 NPC 콘텐츠 개발
- Git Action을 이용한 빌드 자동화 가이드



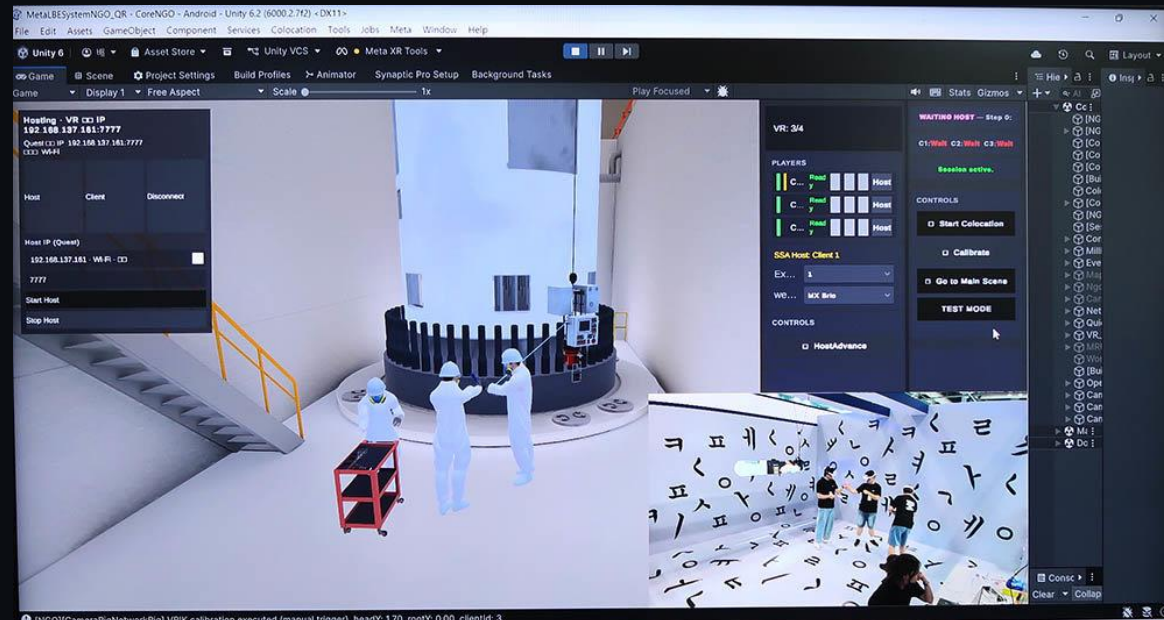
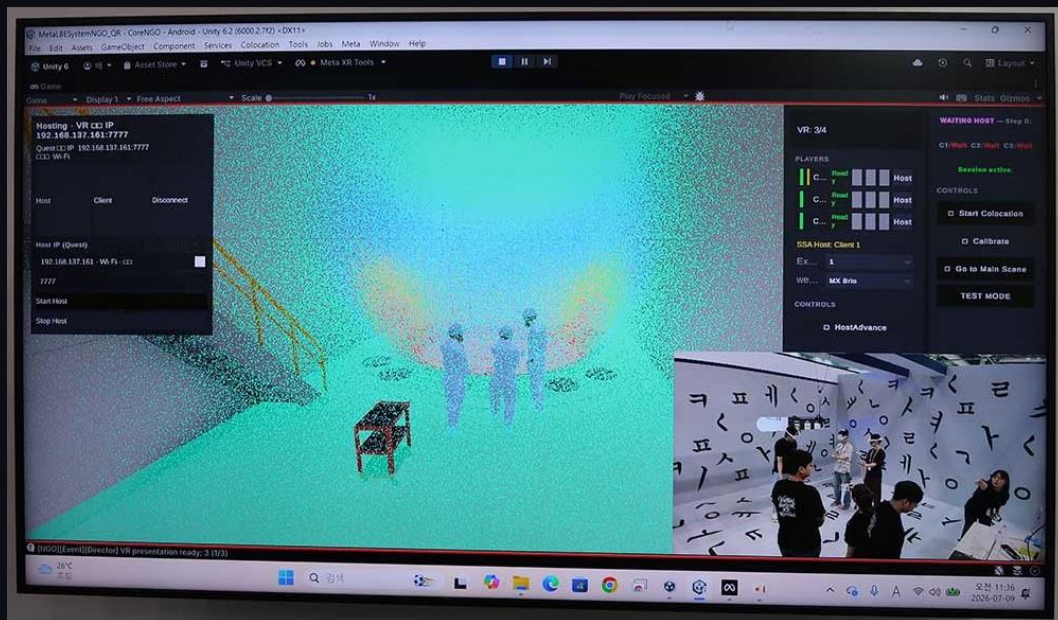


2026.06 ~ 재직중

라이크코퍼레이션

담당업무

- Unity6 VR 콘텐츠 개발





참여 프로젝트 및 목차

프로젝트별 핵심 태그



#멀티플레이#2D대전격투#Unity
#인게임구현#아웃게임구현
#Firebase연동#뒤끝연동#PhotonPUN2



#멀티플레이#PVPVE_3D_익스트랙션
#Unity
#BotAI(BT)구현#매치메이킹 구현
#아웃게임구현
#Firebase연동#PhotonPUN2



#싱글플레이#ARPG#Unity
#인게임구현#Addressable



#싱글플레이#1인프로젝트
#모바일방치형RPG#Unity

GunPowder

팀프로젝트 - 2D 대전 격투 · 실시간 멀티플레이



TNT
GAMES

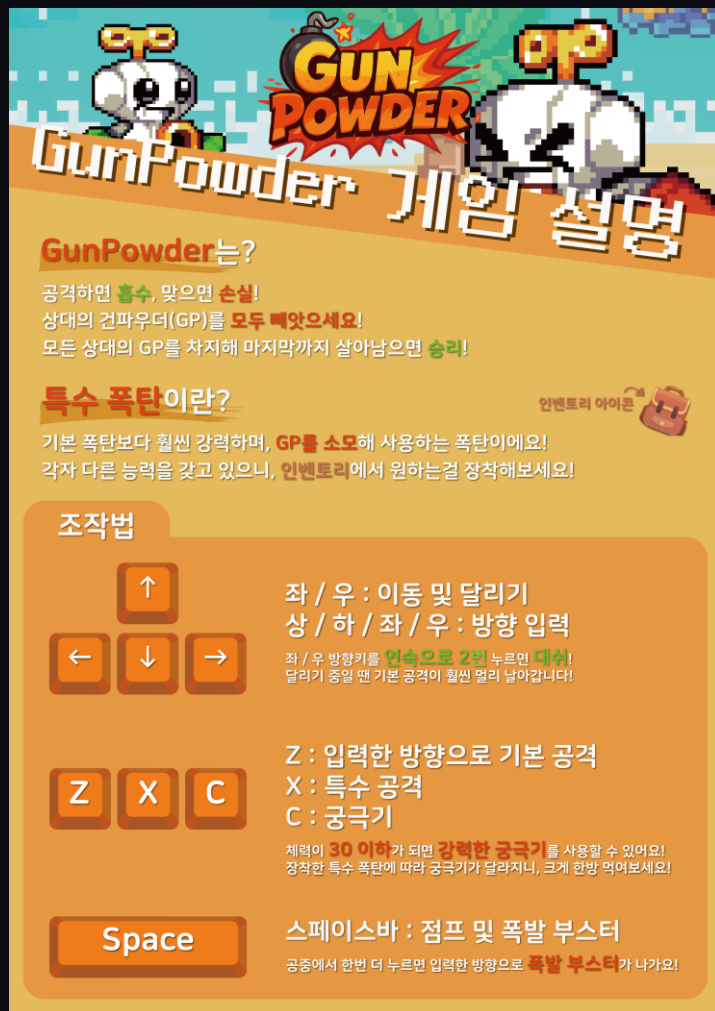
수상 및 공모전

유쾌하고 역동적인 한 판! 우수상 「GunPowder」

- 제 2회 컴투스 글로벌 게임개발 공모전 컴:온 우수상 수상 (2026.04.09)
- 2025 G-STAR 인디쇼케이스 참여 (2025. 11. 13 - 2025. 11. 16)
- 스마일게이트 인디게임 프로토타이핑 챌린지 2기 (2025.11. 1 ~ 2025.11. 22)
- 비버락스 2025 온라인 전시 참여 (2025.12.1 ~ 2025.12.14)



소개



사용 기술

- Unity 6
- C#
- Photon PUN2
- DOTween, (애니메이션 및 트윈 처리)
- 뒤끝 (유저 관리, csv 관리, 채팅)
- Firebase Auth (로그인 인증)
- Firestore (데이터 관리)
- Suno.ai (음악 생성)

- GunPowder는 PUN2 기반 실시간 멀티플레이어 2D 플랫폼 대전 게임입니다.
- 적을 공격하고 GP(건파우더)를 빼앗아 폭탄을 강화하고 더욱 다채롭게 폭탄을 싸보세요!
- 개발 기간: 2025.09.22 - 개발 진행중

게임 소개 영상



YouTube **바로가기** >



<https://youtu.be/O5lymjGapmk?si=fLcywxuDfMwzo5qX>

기여도 및 역할

채팅 시스템 구현

- 채팅 시스템 구현
- 뒤끝 Chat을 이용해 인게임 내 인스턴스 채팅방 구현
- 플레이어의 스킨에 따라 플레이어 아이콘이 변함
- json으로 채팅과 플레이어 스킨 정보를 주고받음
- 채팅 팝업 구현



기여도 및 역할

플레이어 / 카트리지

- 이전 팀원의 작업물에 뒤이어, 새로운 시스템을 위해 리팩토링 및 시스템 추가
- 피격 & 낙백 수정 및 GP(건파우더) 재화용으로 변경, HP 추가
- 플레이어 이벤트 매니저 구현
- 추후 추가될 카트리지를 위해 플레이어 이벤트 매니저를 두어 관리
- 궁극기 게이지 방식 변경, 반동 제어 수정, 피격 낙백 수정
- 기존 상수에서 뒤끝으로부터 csv를 불러와 각 폭탄별 일괄 적용
- 신규 시스템, 플레이어가 선택한 카트리지(증강) 추가 하는 기능 구현
- 카트리지 구현



기여도 및 역할

허리업 시스템 / 기믹

- 허리업 시스템
 - > 게임 내 시간이 30초 남았을 때 랜덤 기믹 발동과 함께 배경음악을 1.2배 가속 구현
- 맵이 배치되는 형식의 기믹 개발
- 기존 기믹 통합과 리팩토링으로 기믹 모듈화 구현



기여도 및 역할

친구, 파티 시스템



- 친구, 파티 시스템
- Firebase에 UUID로 친구 목록을 하다가 뒤끝으로 관리하기 위해 수정
- 친구 추가 / 삭제 / 요청
- 파티 초대 / 추방 / 수락 / 거절

문제 해결 사례

GunPowder

채팅방 중복 접속 및 초기화 타이밍 오류

- 문제:
 - 빠르게 씬을 전환하거나 재접속할 때 같은 채팅 채널에 중복 접속되어 메시지가 2번 표시되거나, UIChatManager의 초기화가 완료되기 전에 호출되어 null 발생
- 해결:
 - 초기화 흐름을 씬 로드 순서와 무관하게 동작하도록 재설계

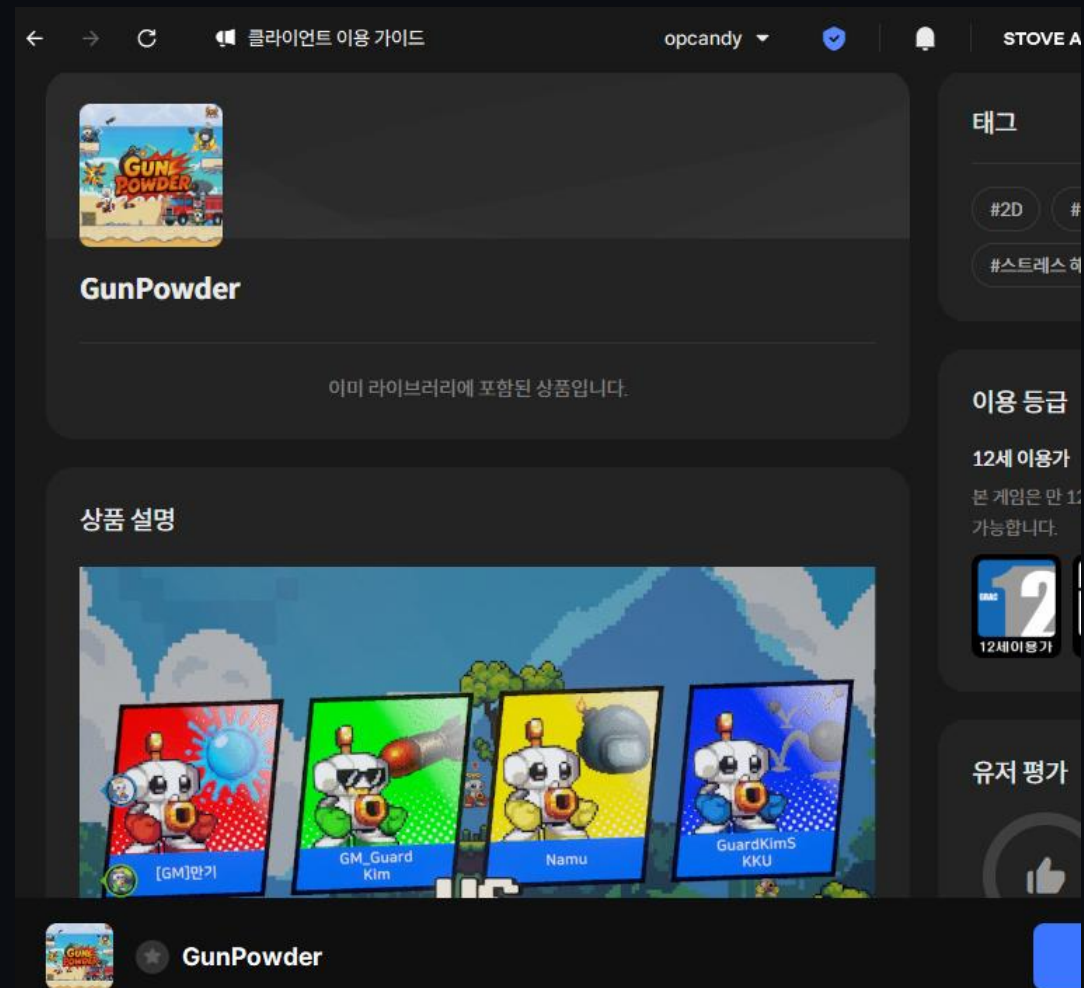
채팅 조작 문제

- 문제 :
 - 채팅하는 유저의 스킨 정보 \$\$DATA\$\$, !partyinvite, 같은 구분자를 채팅 문자열에 함께 송, 수신해서 유저가 조작 할 수 있는 취약점 발견
- 원인 :
 - 빠른 개발을 위해 문자열로 송수신 하던 부분을 채팅 확장(귓속말, 파티초대, 친구초대 등)을 위해 리팩토링
- 해결 :
 - 단순 문자열로 전달하던 부분을 JSON으로 교체, type필드로 메시지 구조를 분리

향후 개발 방향

현재 진행 중인 작업

- GP를 소모하여 얻을 수 있는 증강 시스템(카트리지)을 위한 리팩토링 진행중
- 피격·넉백 시스템 개선: 넉백 방향/세기 계산 로직 고도화 진행 중
- HP/GP 시스템 분리: 체력과 건파우더 게이지 완전 분리 마무리 단계
- 친구, 파티, 채팅 등 커뮤니티 시스템
- 친구 초대, 수락, 삭제, 보류 시스템
- 파티 초대, 수락, 삭제, 보류 시스템
- 로비 및 대기실 채팅 시스템
- 파티 채팅, 친구 채팅 등



협업

Github / Notion

- GitHub 레포를 통해 팀원 간 코드 리뷰와 관리가 이루어졌습니다.
- 개인 작업 브랜치에서 개발 후 Main으로 PR을 생성하여 머지하는 방식으로 운용했습니다.
- G-STAR 전시회 참가 당시 G-Star Build, 긴급 수정 시에는 hotfix 브랜치를 별도 운용했습니다.
- Notion을 통해 작업자들 간의 업무 분담과 스케줄을 관리하였습니다.



외부 전시 및 챌린지

GunPowder



- 2025 G-STAR 인디쇼케이스 참여
(2025. 11. 13 - 2025. 11. 16)
- 스마일게이트 인디게임
프로토타이핑 챌린지 2기
(2025.11. 1 ~ 2025.11. 22)
- 비버락스 2025 온라인 전시 참여
(2025.12.1 ~ 2025.12.14)

CRAVANT

팀프로젝트 - PVPVE 3D 익스트랙션 · Unity



소개 및 영상

<https://github.com/TrollsBrigade/CRAVANT-Proto>



YouTube **바로가기** >



- CRAVANT는 몬스터와 보스, 그리고 플레이어와 치열한 경쟁 속에서 자원을 지배하고 끝까지 살아남는 던전 탐험 RPG 입니다.
- 개발 기간: 2025.07.15 - 2025.08.20

<https://youtu.be/qjadgN9n8Js?si=IUcbqEmU9MugPWIs>

사용 기술

CRAVANT

- Unity 6
- C#
- Photon PUN2
- DOTween, (애니메이션 및 트윈 처리)
- Firebase Auth (로그인 인증)
- Firestore (데이터 관리)
- Suno.ai (음악 생성)

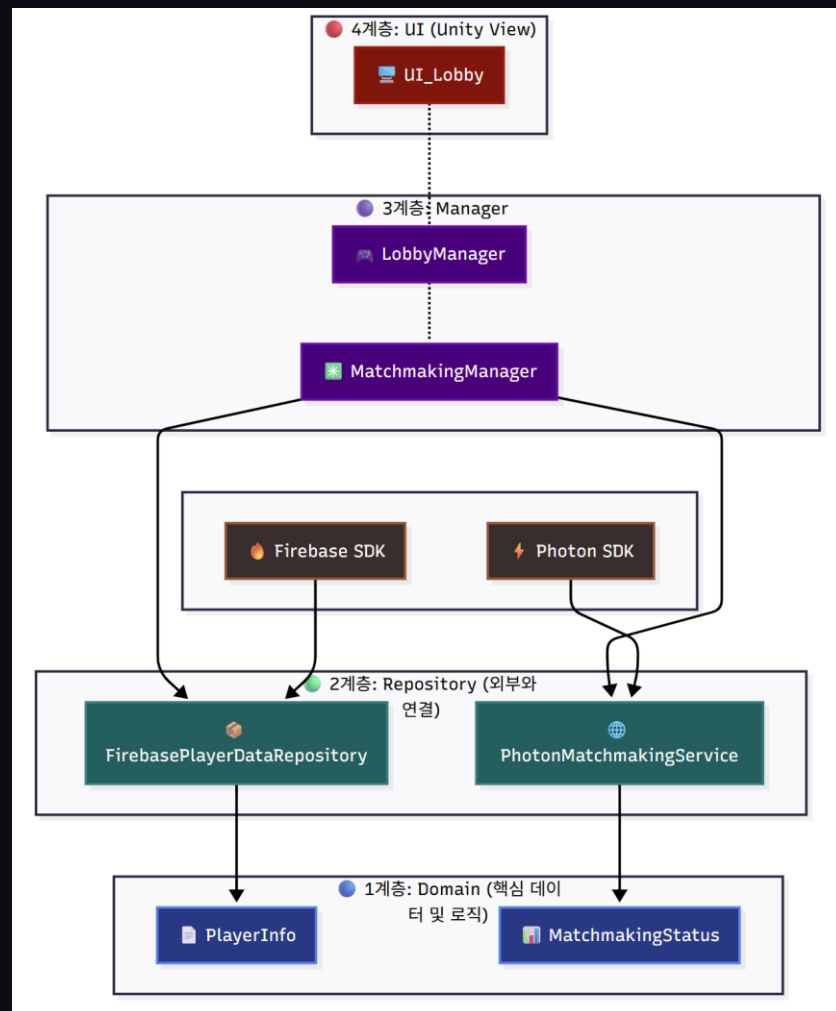


기여도 및 역할

CRAVANT

1. Photon 멀티플레이어 네트워크 시스템 전체 구축

- PhotonServerManager 설계 및 구현
- 솔로큐/다인큐(파티큐) 매치메이킹 시스템 완성
- Firebase UID ↔ PhotonPlayer.UserId 연동
- MasterClient 전환 로직 구현
- 팀 분배 시스템 (같은 파티 → 같은 팀 배정)
- Room 관리 (MaxPlayers 설정, Room 상태 관리)

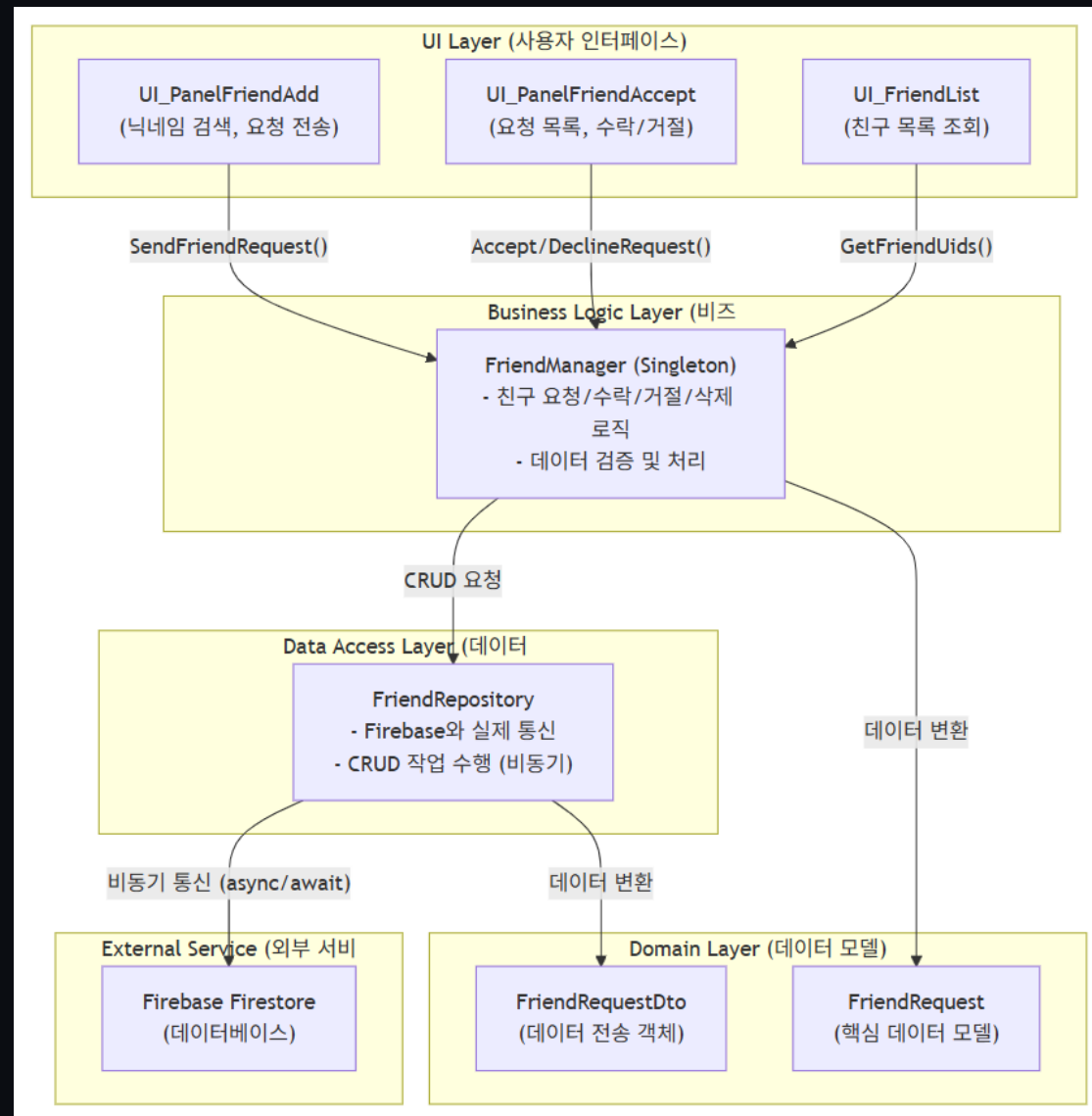


기여도 및 역할

CRAVANT

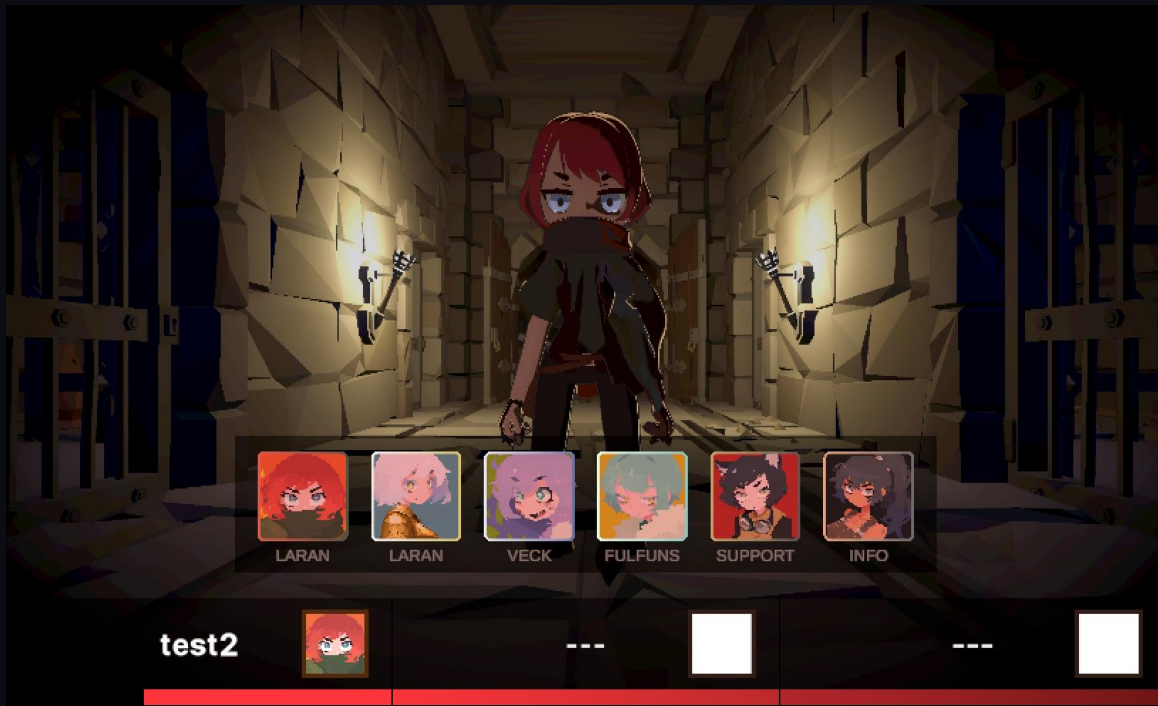
2. 소셜/친구 시스템

- Firestore 기반 친구 기능 전체 구현 (검색, 신청, 수락)
- 친구 파티 초대 → 파티 매치메이킹까지의 전체 플로우
- 유저 검색 기능 (Firestore 구조 설계 포함)
- 로비 UI와 친구 시스템 연동



기여도 및 역할

CRAVANT



3. 매치메이킹, 캐릭터 선택 시스템

- Photon PUN2 기반 솔로 매치메이킹 구현
- 15인 매칭 후 3명씩 5개 팀 자동 분배 로직 구현
- 파티 우선 팀 배정 방식의 파티 매치메이킹 구조 확장
- Room Property 소실로 인한 파티 정보 유실 문제를 Photon Chat으로 보완
- 인원 부족 대응을 위한 봇 포함 매치메이킹 구조 기획
- 캐릭터 선택 시스템 (중복 방지, 타임아웃 자동 선택, 동기화)
- Check-Then-Act 방식으로, Race Condition 해소 후 캐릭터 선택 진행
- PopupManager 기반 UI 관리 체계
- 비동기 씬 로드 전환

기여도 및 역할

CRAVANT

4. Bot AI 시스템 (Behavior Tree)

- BotPlayer 설계 및 BotManager 구현
- Behavior Tree(BT) 기반 Bot AI (Follow, 공격, 무기 사용)
- 봇모드(1인 모드) 완성 — 솔로 플레이 가능한 상태 까지
- Bot 프리팹 생성, Animator 연동, 봇용 무기 생성



기여도 및 역할

CRAVANT

5. 인게임 시스템 기여

- MonsterSpawner / EnemySpawn 시스템 (MasterClient 기반)
- EnemyProjectile(원거리 공격) 구현
- DamageFloaterManager (DamageNumbersPro 연동, 천 단위 콤마 포맷)
- 킬로그 시스템 버그 수정
- TeamManager 책임 분리 리팩토링
- 팀 Indicator / 팀 UI 동기화
- Area 이름 표시 / 플레이타임 표시
- CameraShake, 사망 시 카메라 전환, 미니맵 카메라

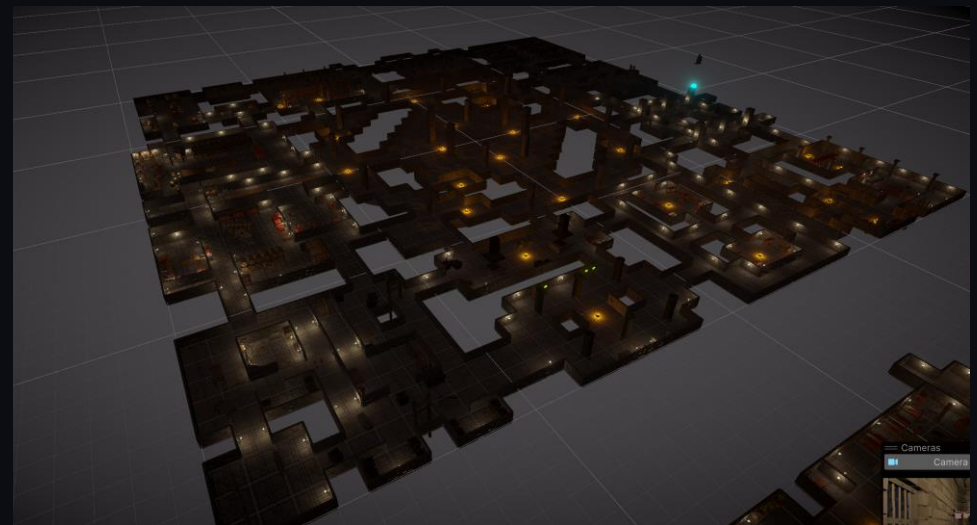


기여도 및 역할

CRAVANT

6. 폴리싱 & UX

- UI_Touchbounce (버튼 인터랙션 애니메이션)
- 사운드 시스템 추가 및 관리
- 인게임 Noise 효과
- 맵 이동 시 Progress Bar
- 매치 결과 화면 구현



문제 해결 사례

CRAVANT

Photon Race Condition 문제

- 문제:
파티 매치메이킹에서 같은 파티는 같은 팀으로 지정되어야 하지만, 다른 팀에 지정되는 현상

- 원인:
Photon의 SetCustomProperties는 호출 즉시 완료되지 않고,
다른 클라이언트들에게 전파되기까지 지연이 있다.
팀 지정 로직이 동기화 되기 전에 즉시 실행되어 일어남

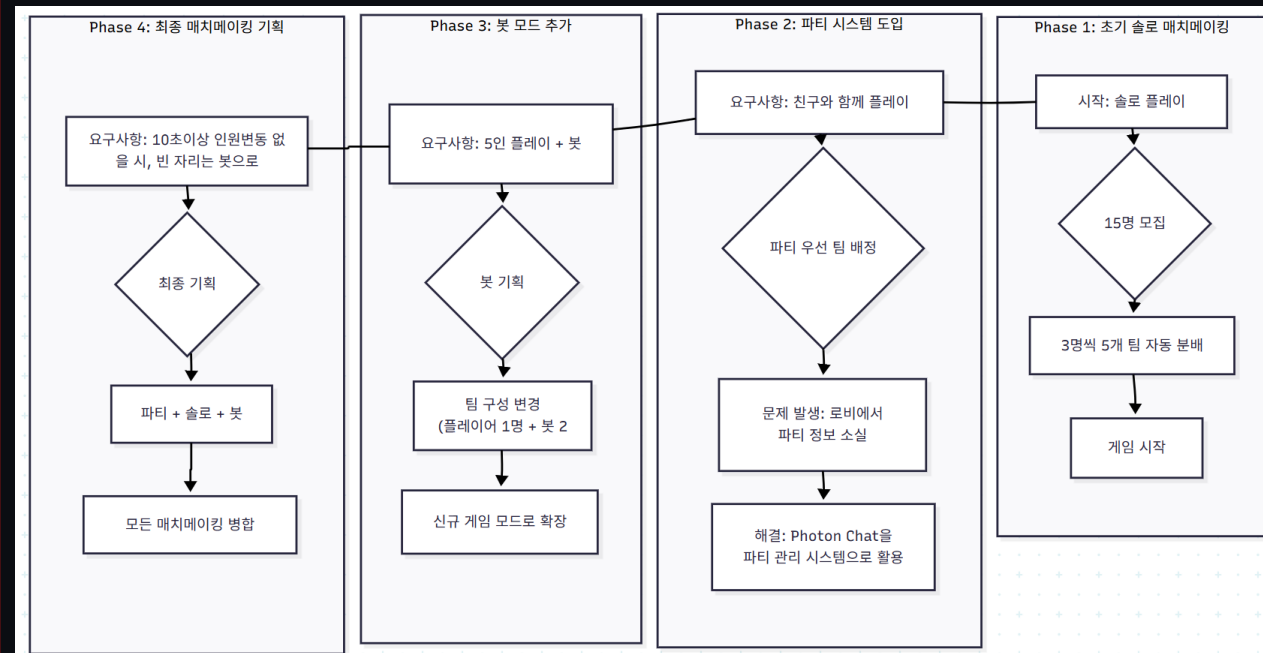
- 해결:
isReadyForTeamAssignment라는 별도의 Condition Check 프로퍼티 추가
MasterClient는 모든 플레이어의 Condition을 확인하고 팀 지정 시작

Bot 스폰 순서 의존성 문제

- 설계 의도:
봇은 플레이어의 데이터에 의존한다(팀 정보, 위치, FollowTarget)

- 문제:
플레이어가 스폰이 끝나기 전에 봇을 스폰하면 참조할 대상이 없어서 null 에러

- 해결:
캐릭터 선택 완료 → 플레이어 Spawn 완료 대기 → 그 후 봇 스폰 시작
스폰 순서 보장을 위해 주기적으로 방의 전체 플레이어의 수와 스폰된 플레이어의 수를 탐색



문제 해결 사례

CRAVANT

C# event 키워드와 캡슐화 - BotBehaviour 상속 문제

- 문제:
CharacterBehaviour에 event Action OnDeath가 선언되어 있고, 이를 상속한 BotBehaviour에서 OnDeath?.Invoke(this)를 호출하니 문제가 발생함
- 원인:
event 키워드는 선언한 클래스 내부에서만 Invoke 가능
- 해결:
CharacterBehaviour에 protected virtual void OnDeathInvoke() 함수를 만들어 자식 BotBehaviour에서 override해서 호출하는 방식으로 해결
- event 키워드를 제거해도 되지만, 캡슐화를 유지하기 위해 함수로 래핑을 선택함.
- 협업:
GitHub 레포를 통해 팀원 간 코드 리뷰와 관리가 이루어졌습니다.
Notion과 Slack을 통해 팀원간 진행도 동기화와 업무 분배가 이루어졌습니다.

최종프로젝트-미르

메시지 캔버스 추가 파일 +

- VFX 추가 - ADD 완료 (redcalypso)

8월 12일 화요일

4. 현재 프로젝트 진행 상황

전체 진행률: 약 65% (완료된 작업 기준)

- Sprint 1: 완료 ✓
- Sprint 2: 완료 ✓
- Sprint 3: 진행중 (70% 완료)
- 보스 몬스터 시스템: 지연 중 ⚠
- 스킬 시스템: 기본 구현 완료, 인디케이터 작업중

5. 현재 프로젝트 구조

아키텍처 & 디자인 패턴:

- 플랫폼: Unity 기반 멀티플레이어 게임
- 네트워크: Photon 기반 매치메이킹 및 실시간 동기화
- 데이터베이스: Firebase Firestore (스킬/캐릭터 데이터)
- 디자인 패턴: Factory Pattern (캐릭터/스킬), Observer Pattern (UI 업데이트)
- 시스템 구조: 모듈화된 스킬 시스템, 상태 기반 캐릭터 관리

6. 주의해야할 점

- ⚠ 중요 이슈사항:
 - 보스 몬스터 패턴 구현 지연 - Sprint 3 일정에 영향 가능성
 - VFX/SFX 리소스 통합 작업 필요
 - 네트워크 동기화 안정성 지속 점검 필요
 - UI 최적화 및 사용자 경험 개선 필요

다음 마일스톤: Sprint 3 완료 목표일 - 8월 말

경호 오전 10:12
Bot 리팩토링 및 버그 수정

수민 오후 5:52
아이템 파밍 레이스 컨디션 해결 / 플레이어 HUD 추가 구현 / 아이템 박스 스폰 및 아이템 삽입 로직 기획에 맞게 구현

미르 오후 5:52
아이템박스 및 박스 스폰 시스템 제작

아이템 - 아이템박스

HealthBox | WeaponBox | ArmorBox 로 구분

1차 → 어떤 아이템 박스를 스폰할 지 선택

Cyberfunk Rider

팀프로젝트 - 사이버펑크 세계관 3D 액션 RPG

소개 및 영상

<https://github.com/AByeong/CyberPunkRiderWaifu>



YouTube **바로가기** >



- Cyberfunk Rider는 사이버펑크 세계관의 3D 액션 RPG 게임입니다.
- 개발 기간: 2025.05.15 - 2025.06.5(22일)

<https://youtu.be/9pMLGwb90QE?si=hlgWiBXIXw2y4DeC>

사용 기술

Cyberfunk Rider

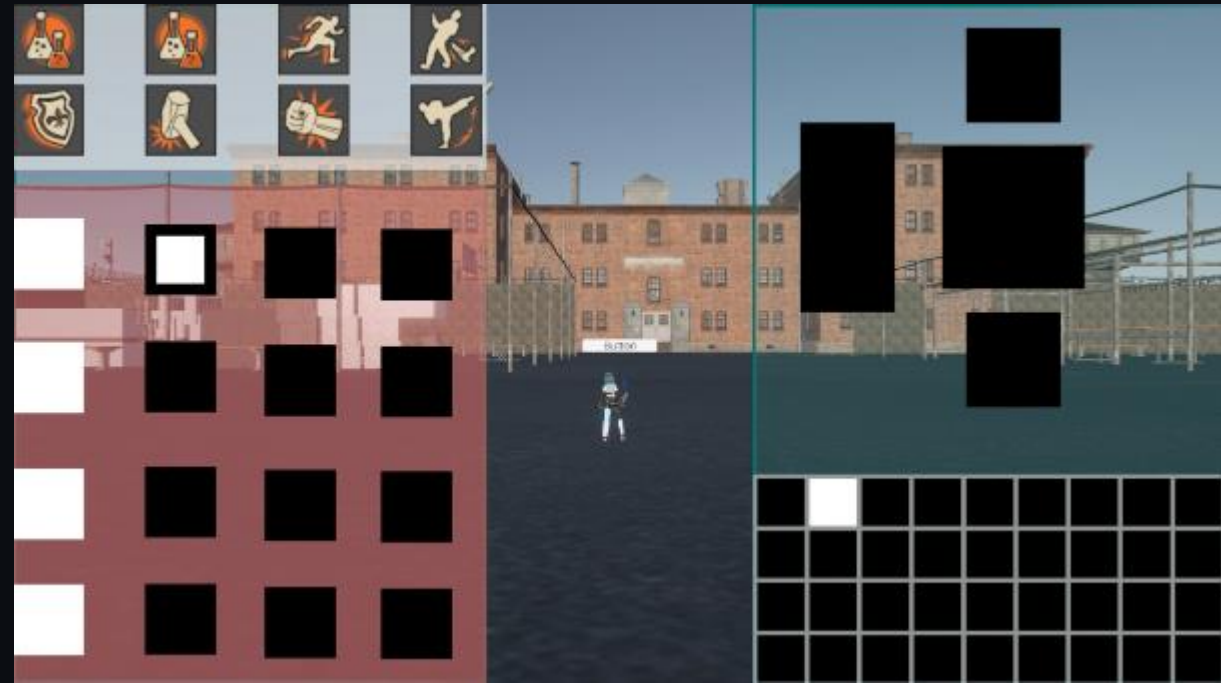
- Unity 6
- C#
- Cinemachine
- Addressables
- UniVRM
- DOTween
- Suno.ai(음악 생성)



기여도 및 역할

인벤토리 시스템

- 인벤토리 시스템: 아이템 CRUD, 슬롯 관리, 드래그&드롭 스왑, 세이브/로드
- 싱글톤 + Observer 패턴으로 전역 인벤토리 상태 관리. 이벤트를 통해 UI와 플레이어 스탯 갱신 로직을 느슨하게 결합
- List기반 런타임 저장소와 데이터 저장을 위한 List 별도 운영
- 장착 아이템만 필터링해 스탯 시스템에 전달
- 20슬롯 용량 제한을 아이템 획득 시점에 검증



기여도 및 역할

장비 시스템

- 장비 시스템: Decorator기반 스탯 시스템, 장착/해제, 세트 아이템 효과
- Decorator로 장비 장착 스탯을 런타임 동적 조합. 인터페이스를 래핑하며 누적
- 구조: Stat(기본) → StatModifierDecorator(장비별) → CriticalStrikeDecorator(크리티컬)
- 장착/해제 시 이벤트 → RefreshStat에서 최종 스탯 재 계산
- EquipmentType은 비트 플래그로 세트 효과 확장 대비
- 희귀도별 스탯 범위 분리 (Normal/Rare/Unique)
- Unique 장비는 세트 효과 + CubeStat 3개(추가 옵션) 랜덤 생성



기여도 및 역할

스킬, 칩 시스템

- 스킬, 칩 시스템: 스킬 매니저, 칩 슬롯, 쿨타임 관리, VFX 연동
- 스킬별 확장 슬롯: 칩 장착/해제 시 해당 스킬만 갱신
- 스킬 사거리/쿨타임을 장비와 독립적으로 조정
- 칩 슬롯: 빈 슬롯일 때 Baselcon 표시



기여도 및 역할

아이템 드롭 시스템

- 아이템 드롭 시스템: 등급별 랜덤 드롭, 오브젝트 풀링, DOTween 연출
- ObjectPool + Coroutine+ DOTween 조합으로 GameFeel 연출. 몬스터가 드롭 아이템 정보를 넘기면 0.3초 간격 순차 드롭
- 등급 확률 테이블: Normal 70% / Rare 25% / Unique 5%로
등급별 VFX 프리팹을 풀에서 획득
- 지면 Raycast 보정: spawn 위치를 구한 뒤 Raycast로 지면 Y를 검출해 허공/지하 드롭 방지
- 포물선 드롭 연출, 드롭 아이템 플레이어 접촉 시 자동 획득 → 파티클/오디오 정리 후 풀로 반환



기여도 및 역할

상점 시스템 / 아이템 생성 시스템

- 상점 시스템: 구매/판매 로직, 드래그 판매, 가격 스케일링
- 구매/판매 탭 UI를 단일 팝업에서 패널 스위칭으로 처리
- 슬롯 타입별 아이템 판매 방식 : 장비/스킬칩은 랜덤 희귀도 생성, 소모품 아이템은 직접 지급
- 누진 가격 시스템 : 구매 시마다 가격 2배씩 상승 → 파밍 밸런스 조절
- 구매 전 2중 검증 골드 부족 / 인벤토리 포화 시 거래 차단
- 판매 슬롯 : 아이템을 드래그해서 떨어뜨리면 골드 획득, 중복 처리 방지
- 아이템 생성 시스템: SO 기반 아이템 팩토리, 랜덤 스탯 생성, 큐브 스탯



기여도 및 역할

드래그 앤 드롭 구현

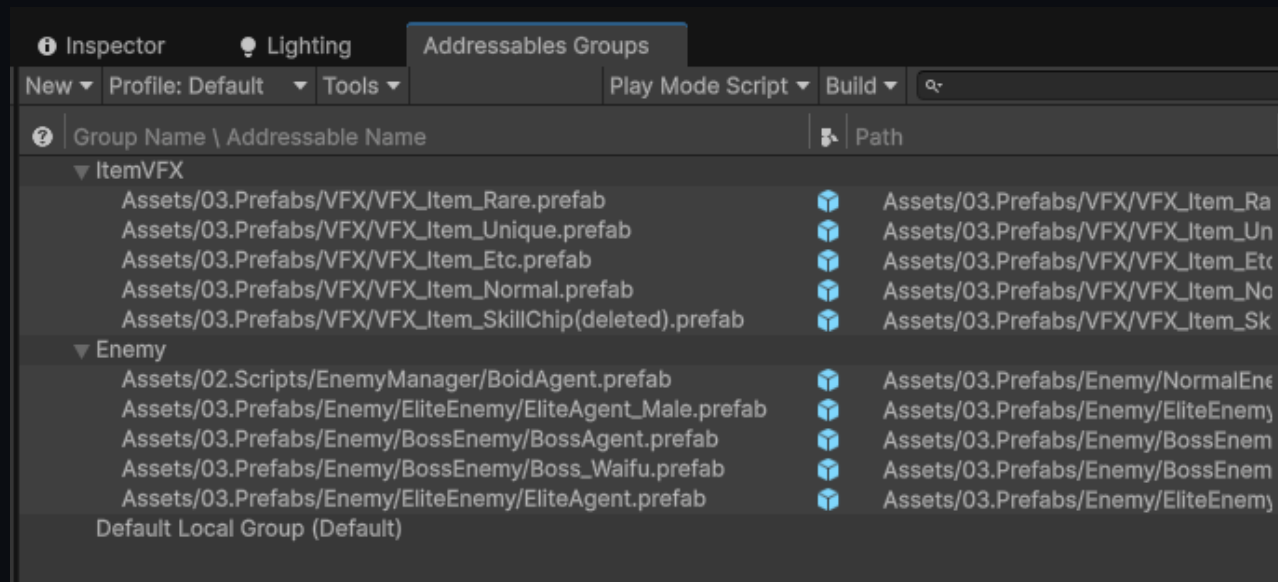
- 드래그 앤 드롭 구현
- 드래그 고스트(반투명 복제)를 마우스 위치에서 추적
원본은 알파 0.5로 표시
- 슬롯 간 이동 규칙 구현 → 잘못된 드롭 차단
- 타입 안전 장비 교체 : 무기 슬롯에 방어구를 장착하면 거부 • 인벤토리 밖으로 드롭 시 아이템 버리기 분기 처리



기여도 및 역할

Addressables 리소스 관리 도입

- Addressables 리소스 관리 도입
 - 드롭 VFX 풀링 (ItemVFX 그룹)
 - 아이템 드롭 시스템이 사용하는 풀을 Addressables + 오브젝트 풀링 조합으로 구현했습니다.
- 인스펙터에 프리팹을 직접 드래그하는 대신 에셋 레퍼런스 리스트를 노출시켜, 풀이 Start()에서 비동기로 에셋을 로드한 뒤 초기 개수만큼 Instantiate하도록 했습니다.
- async/await로 로드 대기 : Task 기반 대기로 콜백 중첩 없이 로딩 흐름을 선형으로 작성
 - 로딩 완료 전에 풀 요청이 오면 null을 반환, 다른 매니저가 먼저 초기화되어도 크래시 없이 처리
 - 씬 전환 시 메모리 누수를 막기 위해 로드한 에셋 핸들을 명시적으로 해제.



협업

아이템 생성 시스템 세부

- 협업:

- GitHub 레포를 통해 팀원 간 코드 리뷰와 관리가 이루어졌습니다.
- Notion을 통해 팀원들간 프로젝트의 진행도 동기화와, 작업분배가 이루어졌습니다.

기본 UI 배치	디자인	UI	● Done	병 병준 강	2025/05/22	High	프로토타입
UI 매니저 제작	코딩	UI	● Done	병 병준 강	2025/05/22	High	프로토타입
팝업 매니저 제작	코딩	UI	● Done	경 경호 김 병 병준 강	2025/05/22	Medium	프로토타입
메인 스테이지 UI	디자인	UI	● Done	병 병준 강	2025/05/23	High	프로토타입
플레이어 스킵 UI	디자인	UI	● Done	병 병준 강	2025/05/23	Medium	프로토타입
플레이어 칩 UI 제작	디자인	플레이어	● Done	경 경호 김 준준 이	2025/05/21	High	프로토타입
-----플레이어-----			● Done				
플레이어 이동 구현	코딩	플레이어	● Done	준준 이	2025/05/19 → 2025/05/20	High	프로토타입
플레이어 공격 구현	코딩	플레이어	● Done	준준 이	2025/05/19 → 2025/05/20	High	프로토타입
콤보 매니저 구현	코딩	플레이어	● Done	준준 이	2025/05/19	High	프로토타입
스킬 1	코딩	플레이어	● Done	준준 이	2025/05/20	High	프로토타입
좌클릭 공격 구현	코딩	플레이어	● Done	준준 이	2025/05/20	High	프로토타입
우클릭 공격 구현	코딩	플레이어	● Done	준준 이	2025/05/20	High	프로토타입
콤보매니저를 활용해 콤보 공격 제작	코딩	플레이어	● Done	준준 이	2025/05/20	High	프로토타입
스킬 연출	디자인	플레이어	● In progress	준준 이	2025/05/22 → 2025/05/23	High	프로토타입
플레이어 콤보 구현	코딩	플레이어	● Done	준준 이	2025/05/20 → 2025/05/21	High	프로토타입
플레이어 스킵 매니저 구현	코딩	플레이어	● Done	경 경호 김	2025/05/19 → 2025/05/21	High	프로토타입
플레이어 장비 시스템 구현	시스템	플레이어	● Done	경 경호 김	2025/05/20 → 2025/05/22	High	프로토타입
플레이어 칩 시스템 구현	시스템	플레이어	● Done	경 경호 김 준준 이	2025/05/22 → 2025/05/23	High	프로토타입
플레이어 칩 효과 스킵과 연동	코딩	플레이어	● Done	경 경호 김 준준 이	2025/05/23	High	프로토타입
-----적-----			● Done				
적 에셋 이동 구현	코딩	몬스터	● Done	병 병준 강	2025/05/19 → 2025/05/21	High	프로토타입
군집 알고리즘 구현	코딩	몬스터	● Done	병 병준 강	2025/05/19 → 2025/05/20	High	프로토타입
적 매니저 구현	시스템	몬스터	● Done	병 병준 강	2025/05/19 → 2025/05/20	High	프로토타입
엘리트 몬스터 이동	코딩	몬스터	● Done	병 병준 강	2025/05/21 → 2025/05/23	High	프로토타입
적 피격 구현	코딩	몬스터	● Done	형 형근 이	2025/05/20 → 2025/05/21	High	프로토타입
적 공격 구현	코딩	몬스터	● Done	형 형근 이	2025/05/20 → 2025/05/21	High	프로토타입
엘리트 몬스터 공격 구현	코딩	몬스터	● Done	형 형근 이	2025/05/22 → 2025/05/24	Medium	프로토타입
엘리트 몬스터 피격 구현	코딩	몬스터	● Done	형 형근 이	2025/05/22 → 2025/05/24	High	프로토타입
보스 몬스터 이동 구현	코딩	몬스터	● Done	병 병준 강	2025/05/23 → 2025/05/24	Medium	프로토타입
보스 몬스터 공격 구현	코딩	몬스터	● Done	병 병준 강	2025/05/22 → 2025/05/23	Low	프로토타입
보스 몬스터 피격 구현	코딩	몬스터	● Done	형 형근 이	2025/05/24 → 2025/05/25	High	프로토타입
-----시스템-----			● Done				
VFX 구하기	디자인	시스템	● Done	병 병준 강 경 경호 김 형 형근 이 준준 이	2025/05/22	Medium	프로토타입
맵 랜덤 생성 알고리즘 적용	코딩	스테이지	● In progress	병 병준 강 형 형근 이	2025/05/21 → 2025/05/22	Medium	프로토타입
스테이지 보상 구현	시스템	스테이지	● Not started	형 형근 이	2025/05/22 → 2025/05/23	High	프로토타입
스테이지 클리어 구현	시스템	스테이지	● Done	병 병준 강	2025/05/22 → 2025/05/23	Medium	프로토타입
-----알파 기간 시작-----			● Done		2025/05/26 → 2025/06/01		

적용된 디자인 패턴

Cyberfunk Rider

패턴	적용 위치	활용 목적
Singleton	InventoryManager SkillManager ItemCreateManager ItemDropManager	전역 상태 관리, 다른 시스템에서의 접근성 보장
Decorator	IStatsProvider StatsDecorator StatModifierDecorator	장비 장착/해제 시 스탯 동적 중첩 및 해제
Observer	Action 이벤트 (OnDataChanged OnEquipChanged OnSkillChange)	UI 자동 동기화, 시스템 간 느슨한 결합
Factory	ItemCreateManager	등급/타입별 아이템 동적 생성
Object Pooling	ItemDropManager ObjectPool	VFX 프리팹 재사용으로 메모리/GC 최적화
Strategy	IWeapon, IEquipable ISkill, IDamageable	무기/장비/스킬의 행위를 인터페이스로 추상화

<https://github.com/AByeong/CyberPunkRiderWaifu>

검물주 키우기

1인 프로젝트 - 모바일 방치형 RPG · Unity



소개

<https://github.com/guardkim/SwordLordMaker>



- 검을 성장시켜 캐릭터를 키우는 모바일 방치형 RPG 게임
- 기간: 2026.01.06 - 2026.01.23(18일)(개발 중단)

핵심 시스템 상세

1. Flying Sword 시스템 — Strategy 패턴 + 수학 궤도

비행 검의 궤도를 Strategy 패턴으로 캡슐화하여, 런타임에 궤도를 자유롭게 전환할 수 있도록 설계했습니다.



궤도 이름	참고한 공식	전투 특성
Adel	리사주(8자) 곡선	유선형으로 움직이는 형태 검의 스폰순서대로 순차적으로 공격함 (준비, 공격, 대기)
Hypo	하이포사이클로이드 곡선	조금 더 꽃잎 형태에 가까운 곡선 형태 순서에 상관없이 검 마다 공격 쿨타임을 가짐
Pixel	렘니스케이트 곡선	원형에 가까운 곡선 형태 순서에 상관없이 검 마다 공격 쿨타임을 가짐

핵심 시스템 상세

2. BigInteger 무한 스케일링 시스템

- 방치형 RPG에서 수치가 기하급수적으로 증가하는 문제를 System.Numerics.BigInteger로 해결했습니다.
- 적용 범위: 공격력, HP, Gold, Ruby, 강화 비용 등 게임 내 주요 수치 전반
- 기술적 도전과 해결:
 - BGDatabase가 BigInteger를 미지원 → string ↔ BigInteger 변환 레이어를 Repository에서 처리
- 비율형 보너스(치명타 확률)는 float, 수량형(공격력·체력)은 BigInteger → 하이브리드 연산 설계
- 스테이지 배율 적용 시 float 정밀도 손실 → ×1000 정수 보정 로직으로 정밀도 확보
- CurrencyFormatter로 큰 숫자를 A, B, C... 단위로 가독성 있게 표현했습니다.



핵심 시스템 상세

3. Enemy AI — FSM + Object Pool

- 적 AI를 유한 상태 기계(FSM) 6개 상태로 구성하고, 오브젝트 풀링으로 성능을 최적화했습니다.
- FSM 상태: Idle → Chase → Attack → SkillAttack → Hit → Dead
- NavMeshAgent 기반 경로 탐색 이동
- UnityEngine.Pool.ObjectPool로 일반 몬스터 풀링 → GC 부하 최소화
- 보스는 AoE 스킬 공격을 수행하며, 별도 생명주기(Destroy) 관리
- 풀링 재사용 시 Animator.Rebind()로 애니메이션 상태 완전 초기화



핵심 시스템 상세

4. Currency & Upgrade 시스템

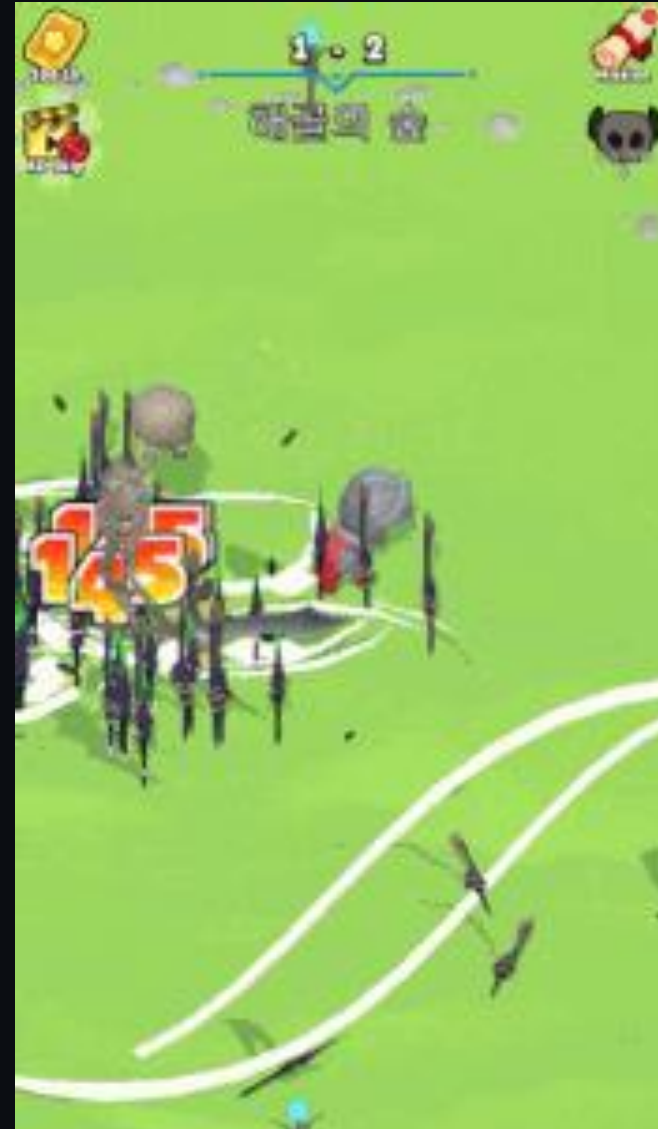
- 재화 관리:
Gold/Ruby를 BigInteger로 관리하며 안전한 연산 메서드를 제공합니다.
60초 간격 자동 저장 + 앱 종료 시 동기 저장으로 데이터 유실 방지.
Ruby는 즉시 저장, Gold는 배치 저장으로 최적화했습니다.
- 강화 시스템:
 $\text{BaseCost} \times (\text{CostMultiplier} ^ \text{Level})$ 지수 비용 곡선을 BigInteger로 연산합니다
강화 상태는 JSON 직렬화로 BGDatabase에 저장합니다.
- Manager 이벤트만 구독하여 비즈니스 로직이 UI에 침투하지 않도록 설계했습니다.
- CurrencyUI는 LateUpdate 패턴으로 매 프레임이 아닌 변경 시에만 갱신합니다.
- RedDot, Popup, Fade 등 재사용 가능한 공통 컴포넌트 구현했습니다.



핵심 시스템 상세

5. Stage 진행 시스템, DamageFloater 시스템

- 스테이지 루프와 플레이어 사망/부활 흐름을 이벤트 체인으로 연결했습니다.
- 게임 루프:
몬스터 무한 스폰 → 일정 수 이상 처치시 보스 입장 가능 →
보스 입장 → 보스 처치 → 다음 스테이지
- 사망/부활 흐름: 사망 → 5초 대기 후 자동 부활 → 해당 스테이지의 1단계로 이동
- 타격감을 높이기 위해 7가지 애니메이션 스타일의 데미지 텍스트 시스템을 구현했습니다.
- Critical 히트 시 별도 색상/크기 적용
- Billboard 회전으로 항상 카메라를 향하도록 처리
- FloaterOption 구조체로 인스펙터에서 연출 파라미터 세밀 조정 가능
- Z-Order 관리로 Z-Fighting 방지



개발을 하며 배운 점

해당 경험을 통해 알게된 점



- 문제 배경
 - 장르 이해 부족 + 초기 과도한 아키텍처 → 기획 변경 비용 증가
- 해결 방법
 - 개발 중반부터 레퍼런스 게임 분석을 병행하여 성장 구조를 재설계하고, 아키텍처 측면에서는 과도한 추상화를 단순화하는 점진적 리팩토링을 진행했습니다.
- 기획이 확정되지 않은 초기에는 빠른 프로토타이핑이 가능한 단순 구조로 시작하는 것이 현실적으로 나은 방안이라는 생각이 들었습니다.
- 또한 개발 전에 동일 장르 레퍼런스를 충분히 분석하여 기획 수정 자체를 줄이는 것이 아키텍처보다 먼저라는 것을 배웠습니다.

감사합니다

김경호 · Client Developer Portfolio

Email kimkyoungho0726@gmail.com

GitHub <https://github.com/guardkim>