

LEE**E**HYOW**O**N

플레이어의 **재미**를 추구하는 게임을 만드는 것이 **목표**인 개발자

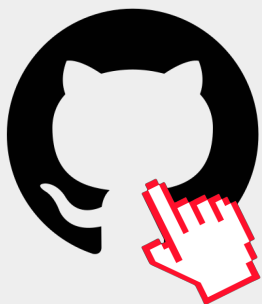


이효원

2002. 08. 13

010-2327-9714

reppy0813@gmail.com



01 EDUCATION

2021. 03 – 2023.02 충남도립대학교 컴퓨터 공학과 졸업 (졸업 학점 4.3/4.5)

02 TRAINING & COURSES

2024. 12 – 2025. 06 청년취업사관학교 – 언리얼 엔진과 게임 개발자 양성 과정

2023. 06 – 2024. 02 한국정보교육원 – 자바 스프링 리엑트를 활용하는 클라우드 풀스택 개발

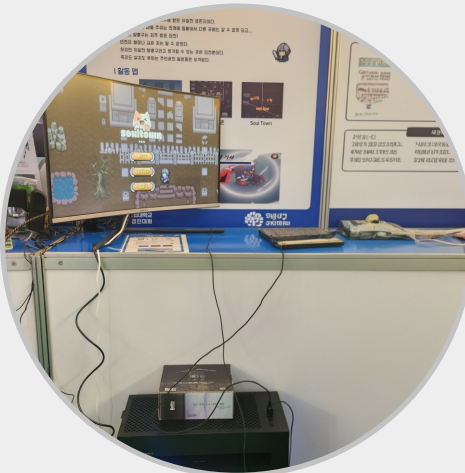
03 COMPETITIONS



메타버스 엑스포 부스 사진

2025. 06 .17
메타버스 엑스포 2025 출품

- Project Aurora 출품
- 200명~ 300명 시연



캡스톤 출품 부스 사진

2023. 12. 23
대학교 캡스톤 출품 (2023)

- 최우수상 수상
- SoulTown 출품
- 40명 ~ 50명 시연

04 CORE SKILLS



05 LEARNING SKILLS



Projects

01 PROJECT AURORA

[Page 04 →](#)

Unreal Endge 기반 창작 3인칭 고난이도 액션RPG

02 SOULTOWN

[Page 13 →](#)

Unity 기반 창작 2d 탑다운 로그라이크 게임

03 테토루와 아이루

[Page 17 →](#)

Unreal Engine기반 몬스터헌터 무페토전 모작 Listen Server 멀티플레이 게임 제작

04 SUSUSUSUPER-NOVA (VR)

[Page 21 →](#)

Unreal Engine기반 Supermargket Together 모작

05 GEEK-LOL

[Page 23 →](#)

React, Spring 제작 롤 OP.GG 홈페이지 모작

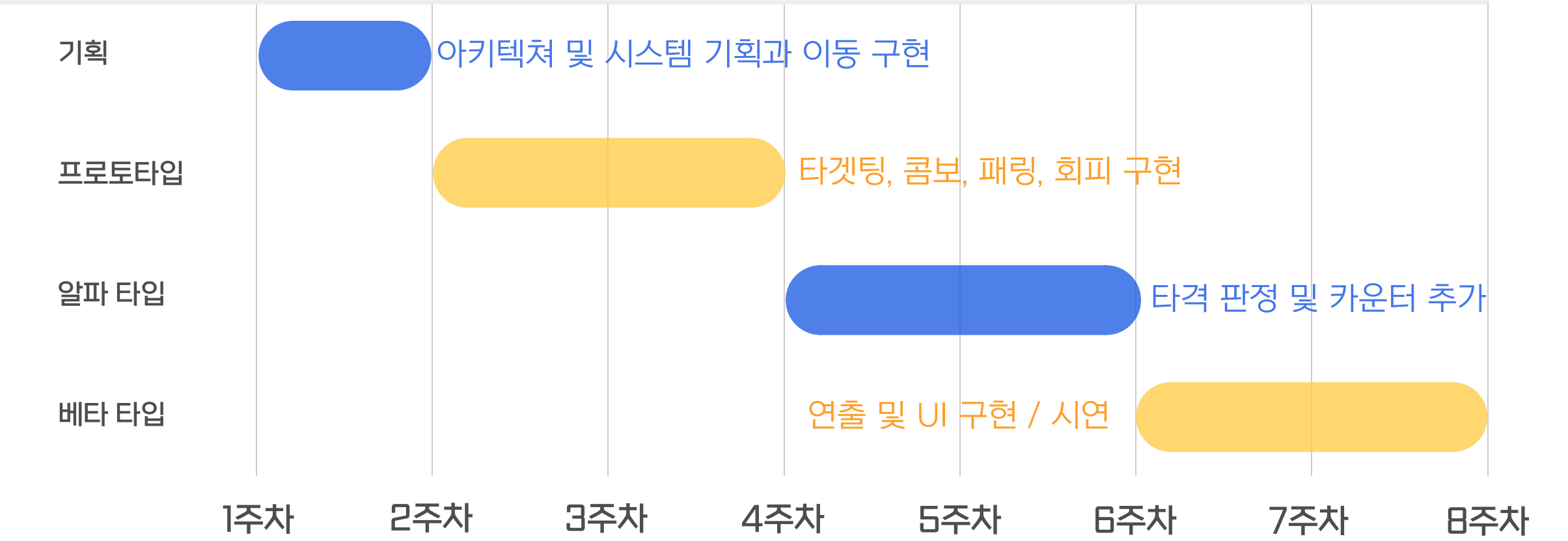
06 OPERATION_DDT - 진행중인 프로젝트

[Page 26 →](#)

액션 슈팅 소울라이크 (PVE, TPS)

프로젝트 소개

장르	UNREAL ENGINE 기반고난이도 액션 RPG
기획 의도	자세 전환을 통한 전술적 선택과 타격 피드백의 강화를 핵심으로 한 전략적 1:1 액션 게임
게임 시스템	패링, 저스트 회피, 카운터 어택, 무기 변경 등 정교한 타이밍 기반의 전투 시스템을 구현
개발 기간	2025. 05. 07 ~ 2025. 06. 17
제작 인원	03명 (클라이언트 개발자 2명, 기획자 1명)
사용 기술	UNREAL ENGINE5 / C++ / GIT / GITHUB
GITHUB	깃 허브 이동하기



본인 구현 Gameplay Systems & Features

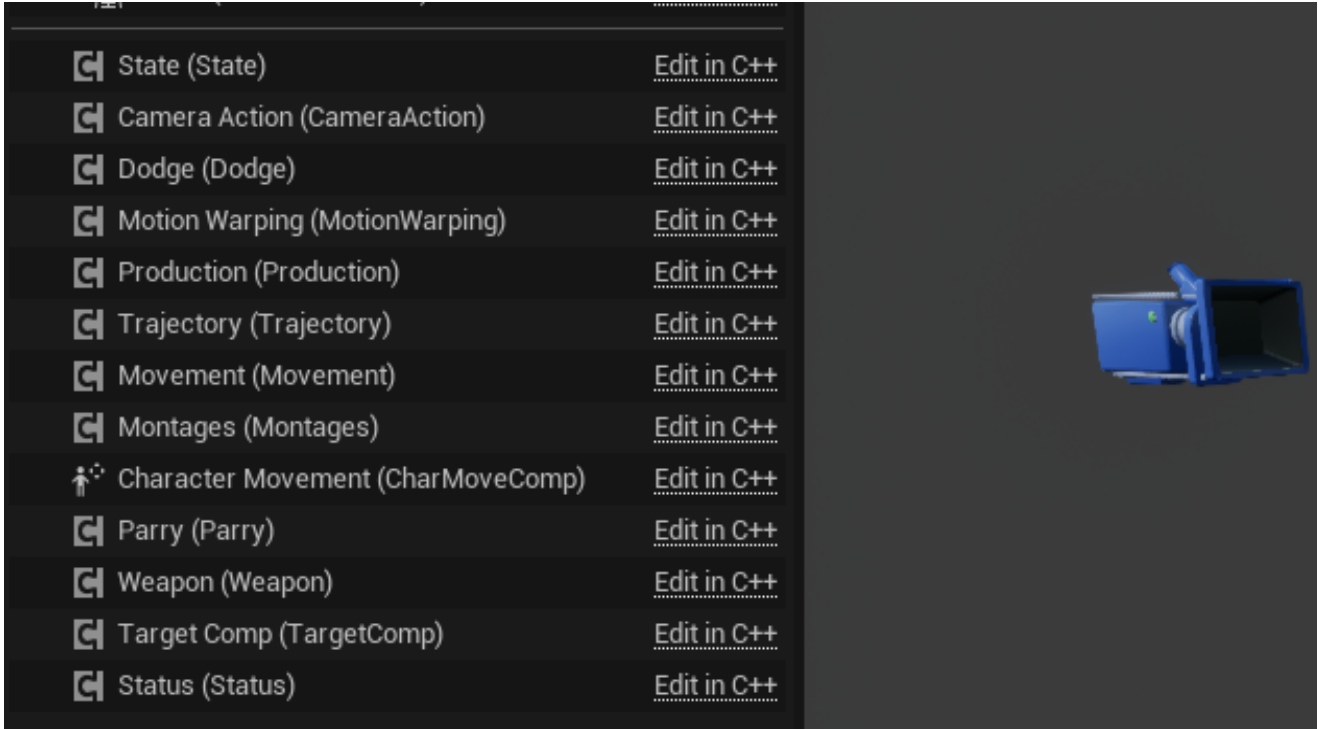
아키텍처	전투·이동·상태·연출 등 기능을 ActorComponent 10종으로 분리 설계 플레이어와 Enemy가 동일 구조를 사용하도록 공용화
무기 시스템	WeaponComponent에서 무기 데이터·Montage·Attachment를 통합 관리 무기별 공격·피격·패링·회피 로직을 데이터 기반으로 처리
타격 판정	소켓 궤적을 Bézier Curve로 보간해 프레임 단위 타격 판정 구현 패링, 저스트 회피, 카운터 공격을 입력 타이밍 기반으로 분기
리액션	사분면 기반 패링/저스트 회피, 성공 시 카운터+Motion Warping 연계
콤보	강·약(LMB/RMB) 분기 및 ComboCount 공유 → 연속 공격 지원
타겟팅	Auto Targeting과 Motion Warping으로 적 위치에 맞춰 공격 위치 보정
이펙트/연출	Socket 좌표를 따라 Niagara Trail 생성으로 검 궤적 이펙트 적용 카메라 액션과 Montage 타이밍을 동기화해 애니메이션 연출 제어
UI 플로우	게임 전반 Intro~Settings까지 이어지는 전체 UI 플로우 설계 전투 HUD(무기 교체, 스탯/보스 HP Bar) 등 진행 상황 시각화 구현



주요 구현 내용

재사용성과 확장성을 고려한 기능 컴포넌트화 설계

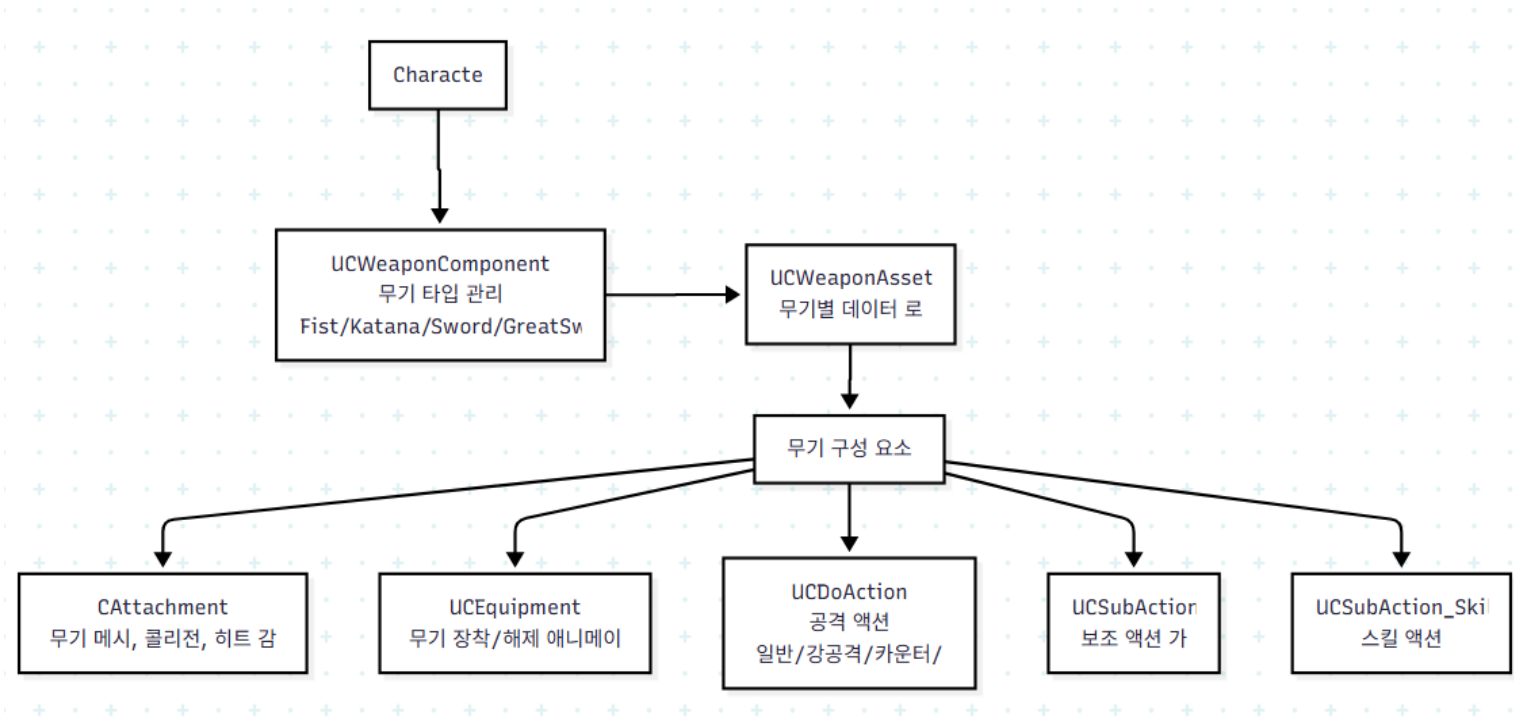
- 도입 배경
 - 이전 프로젝트에서 기능 간 높은 의존성으로 유지보수가 어려웠고, 재사용성 및 확장성 강화를 위해 컴포넌트화 설계를 도입
- 구현 방식 및 역할
 - 모든 전투 및 제어 기능을 개별 Actor Component로 분리하여, 플레이어와 Enemy 모두 동일한 시스템을 재사용 가능하도록 설계
 - 기능 단위로 모듈화함으로써 객체지향 설계 원칙(OOP)의 캡슐화와 확장성을 실현
 - 각 컴포넌트는 특정 기능에만 집중할 수 있도록 단일 책임 원칙(SRP)을 기반으로 설계
- 성과 및 효과
 - 코드 중복 감소 및 유지보수 용이성 향상 및 공통 컴포넌트 재사용으로 개발 생산성 증대



언리얼 에디터 내 플레이어 구조

Weapon Component 기반의 유연한 무기 관리 및 공격/피격 시스템

- 도입 배경
 - 다수의 무기를 효율적으로 관리하고 향후 확장을 용이하게 하고자, 시스템의 종속성은 낮추고 확장성은 높이는 것을 목표로 도입.
- 구현 방식 및 역할
 - UWeaponComponent를 중심으로 무기 장착, 공격 로직을 통합 관리하고 역할을 분리.
 - 무기 데이터(UWeaponData)와 월드에 배치되는 액터(AAttachment)를 분리하여 데이터 기반의 확장성 확보.
 - 무기 액터를 인스턴스로 생성 및 관리하여, 무기별 고유 Montage 및 패링, 회피 등 차별화된 리액션 시스템 적용이 가능하도록 설계.
- 성과 및 효과
 - 기존 코드에 미치는 영향을 최소화하며 안정적으로 시스템을 확장 가능.



WeaponCompoent 플로우 차트

주요 구현 내용

■ Bézier Curve 기반 Line Trace Hit Detection 판정

- 도입 배경
 1. 기존의 무기에 부착된 단일 콜리전(Collision) 판정 방식은 실제 공격 궤적과 차이가 발생. 빠른 속도의 공격에서 프레임 간 터널링 현상이 생겨 정확한 피격 판정에 한계가 있었음.
- 구현 방식 및 역할
 1. 무기 양 끝점에 Socket을 부착하고, 프레임 간 위치를 Bézier Curve로 보간하여 실제 휘두르는 궤적을 예측.
 2. 예측된 궤적을 따라 소켓이 이동한 공간을 다수의 삼각형 면(Triangle Face)으로 재구성하여 판정 영역을 생성.
 3. Bézier Curve로 보정된 궤적에 따라 삼각형 영역 내부에 Line Trace를 실행, 면 단위의 정밀 충돌 판정을 구현
- 성과 및 효과
 1. 프레임 드랍과 관계없이 일관되고 정확한 타격 판정을 보장함. 시각적인 무기 궤적과 실제 피격 판정 영역을 일치시켜 플레이 경험의 현실감을 증대시키고, 최적화된 충돌 감지를 구현함.



Line Trace Hit Detection Debug 시각화

■ Auto Targeting 기반의 공격 방향 자동 조준 시스템

- 도입 배경
 1. 명시적인 록온(Lock-on) 시스템이 주는 UI/UX의 제약을 배제하고, '고스트 오브 쓰시마'와 같이 자연스러운 전투 몰입감을 제공하고자 소프트 타겟팅 시스템을 도입.
- 구현 방식 및 역할
 1. 공격 시작 시, 전방 일정 거리 내 Enemy들을 대상으로 플레이어의 입력 방향과 시야각을 기준으로 계산하여 최적의 타겟을 탐색.
 1. 탐색된 타겟을 향해 공격 방향을 자동으로 보간하여, 사용자의 조작 의도를 해치지 않는 선에서 정확한 공격을 유도.
 2. 플레이어의 달리기 상태, 혹은 일정 시간 비공격 상태가 지속될 경우 타겟팅을 자동으로 해제하여 어그로가 고정되는 현상을 방지.
- 성과 및 효과
 1. 별도의 록온 조작 없이 플레이어의 공격 의도를 지능적으로 보조하고 UI를 간소화하여, 향상된 조작감과 함께 전투 몰입감을 극대화함.



Auto Targeting 적용 화면

주요 구현 내용

■ 타이밍과 방향성 기반의 리액티브 전투 시스템

- 도입 배경
 1. 플레이어의 정밀한 조작과 판단이 전투의 승패를 가르는 고속런 전투 시스템을 목표로 도입.
 2. 적의 공격 방향과 타이밍에 맞춰 정확하게 반응하고, 이를 강력한 반격으로 연결하는 경험을 통해 전투의 긴장감과 성취감을 극대화하고자 함.
- 구현 방식 및 역할
 1. **방향성 패링**: Box Line Trace로 공격 위치를 사분면 판정, 공격이 들어온 방향과 일치하는 패링 Montage를 재생하여 수비 리액션의 직관성과 설득력을 확보.
 1. **Just 회피**: 8프레임의 짧은 판정 시간 내 회피 성공 시, 전용 'Just 회피' Montage를 출력하여 고난도 조작에 대한 즉각적인 시각적 보상을 제공.
 2. **Counter 연계**: 성공적인 패링/회피 직후, Motion Warping이 적용된 카운터 공격으로 연계. 타겟을 정확히 추적하여 공격을 명중시키는 강력한 보상을 통해 '수비 후 반격'의 쾌감을 강화.
- 성과 및 효과
 1. 플레이어의 조작 숙련도가 전투 효율로 직결되는 시스템을 구축하여 도전적인 플레이 경험을 제공.
 2. '수비-회피-반격'으로 이어지는 유기적인 액션 사이클을 통해, 단순한 전투를 넘어선 전략적인 공방의 재미와 높은 몰입감을 선사.



방향성 패링 인게임



패링 이후 카운터 인게임

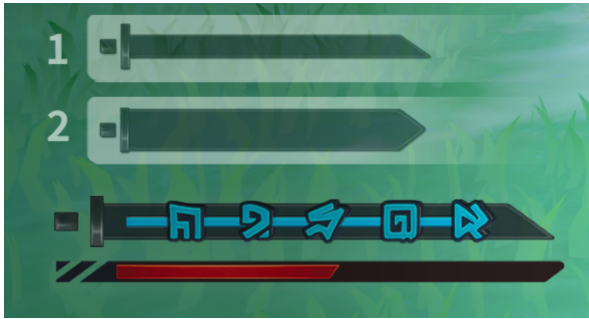
주요 구현 내용 - UI

■ Player State UI : 무기 교체 연출과 스탯 시각화를 결합한 전투 HUD 구성

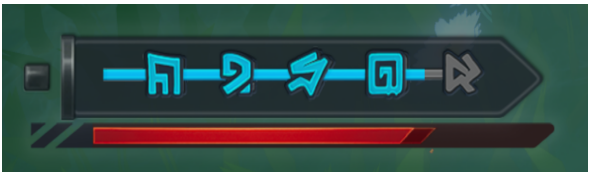
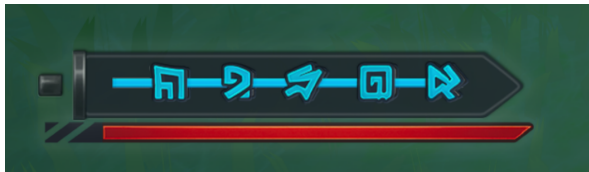
- 무기 교체 파트
 - 도입 배경
 - 1. UI 애니메이션과 캐릭터의 실제 동작(Montage)을 연동하여, 무기 교체 완료 상태를 유저에게 확실하게 피드백하는 세련된 연출을 목표로 설정.
 - 구현 방식 및 역할
 - 1. Tab 키 입력 시 UI 회전 연출과 무기 교체 Montage를 동기화하여 재생.
 - 2. FWidgetAnimationDynamicEvent를 활용하여, UI 애니메이션의 각 단계에 맞춰 캐릭터의 교체 Montage가 정확한 타이밍에 실행되도록 시스템 설계.
 - 성과 및 효과
 - 단순한 기능적 UI를 넘어, Delegate 기반의 동적인 연출을 결합하여 플레이어의 시각적 만족도와 교체 피드백의 명확성을 동시에 확보.
- 플레이어 스탯 파트
 - 도입 배경
 - 1. 피격 손실량과 자원 소모량을 명확히 인지시켜 전략적인 플레이를 유도하고자 UI를 설계.
 - 구현 방식 및 역할
 - 1. HP Bar: 현재 체력(Red Bar)과 피격 대미지량(Dark Bar)을 보여주는 2중 구조로 구성하여, 플레이어가 받은 피해량을 즉각적으로 체감하도록 구현.
 - 2. Mana Bar: 20% 단위로 분할된 디자인으로 구성하여, 스킬 사용 시 소모되는 자원의 양을 시각적으로 명확하게 인지할 수 있도록 설계.
 - 성과 및 효과
 - 1. 플레이어가 자신의 상태를 쉽고 빠르게 파악하여 전투 중 더 나은 전략적 판단을 내릴 수 있도록 지원.



인게임 화면



무기 변경 UI



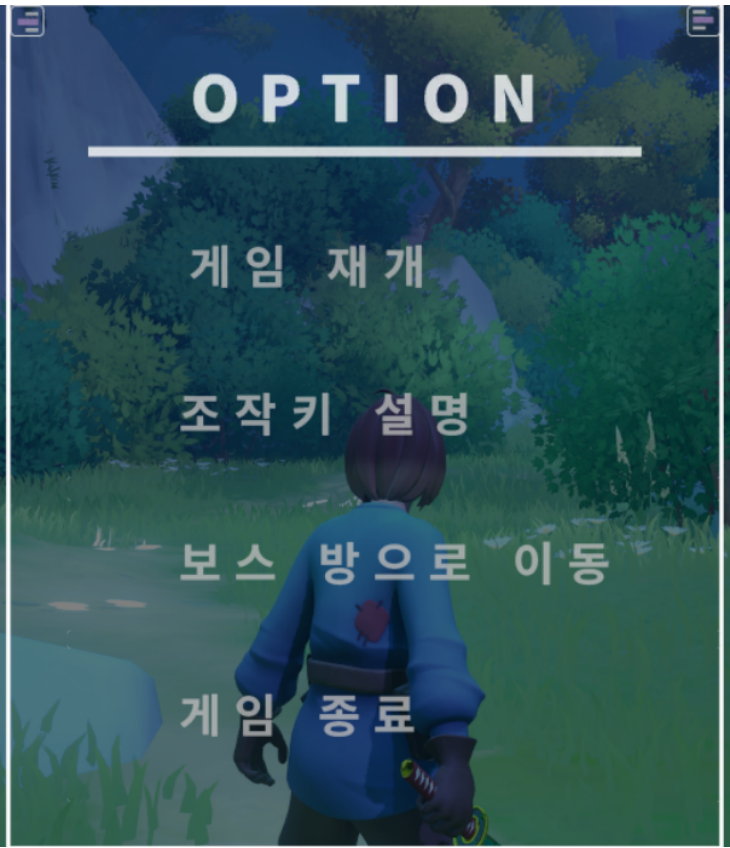
HP 프로그래스 바 UI

주요 구현 내용 - UI

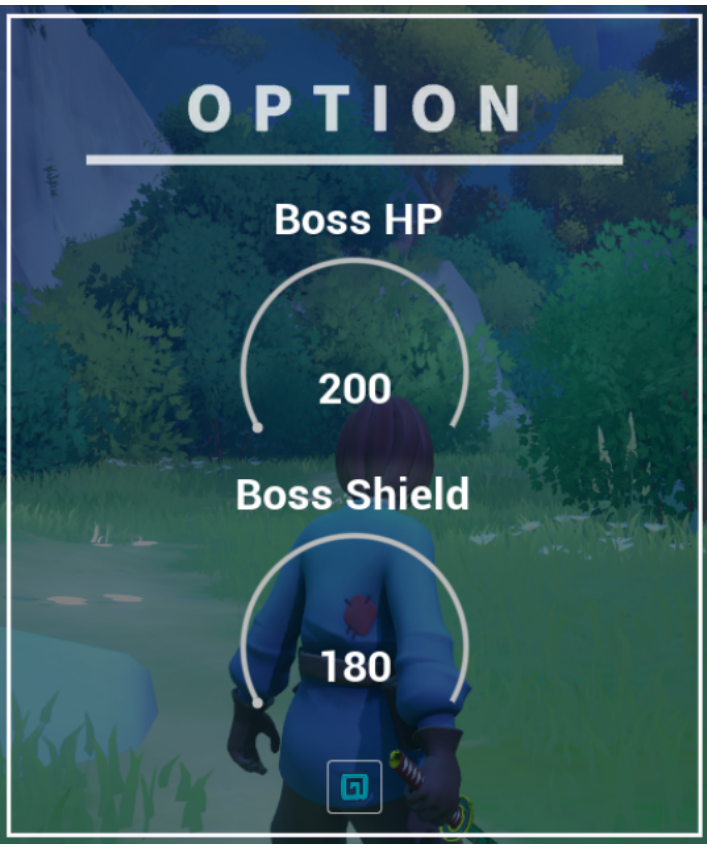
- Settings UI : 입력 모드 전환과 디버그 기능이 포함된 정지 메뉴 시스템
 - 도입 배경
 1. '메타버스 엑스포' 현장 시연 중 실시간 난이도 조절 및 리셋의 필요성과 플레이어의 기본 편의 기능을 통합 관리하기 위해 개발자 모드가 내장된 메뉴 시스템을 구축.
 - 구현 방식 및 역할
 1. SetInputMode (UI/Game Only) 전환 로직으로 메뉴 활성화 시점의 사용자 입력을 명확하게 제어.
 2. Esc 키를 통한 직관적인 메뉴 네비게이션과 함께, 플레이어 편의 기능 및 개발자용 디버그 툴(게임 리셋, 난이도 조절)을 통합 구성.
 - 성과 및 효과
 1. 안정적인 InputMode 제어를 통해 플레이어에게는 끊김 없는 플레이 경험을 제공하고, 개발자에게는 효율적인 실시간 디버그 환경을 제공하여 시연 및 테스트의 완성도를 높임.

■ 절차적 UI 전환을 통한 몰입형 게임 인트로 시퀀스 구현

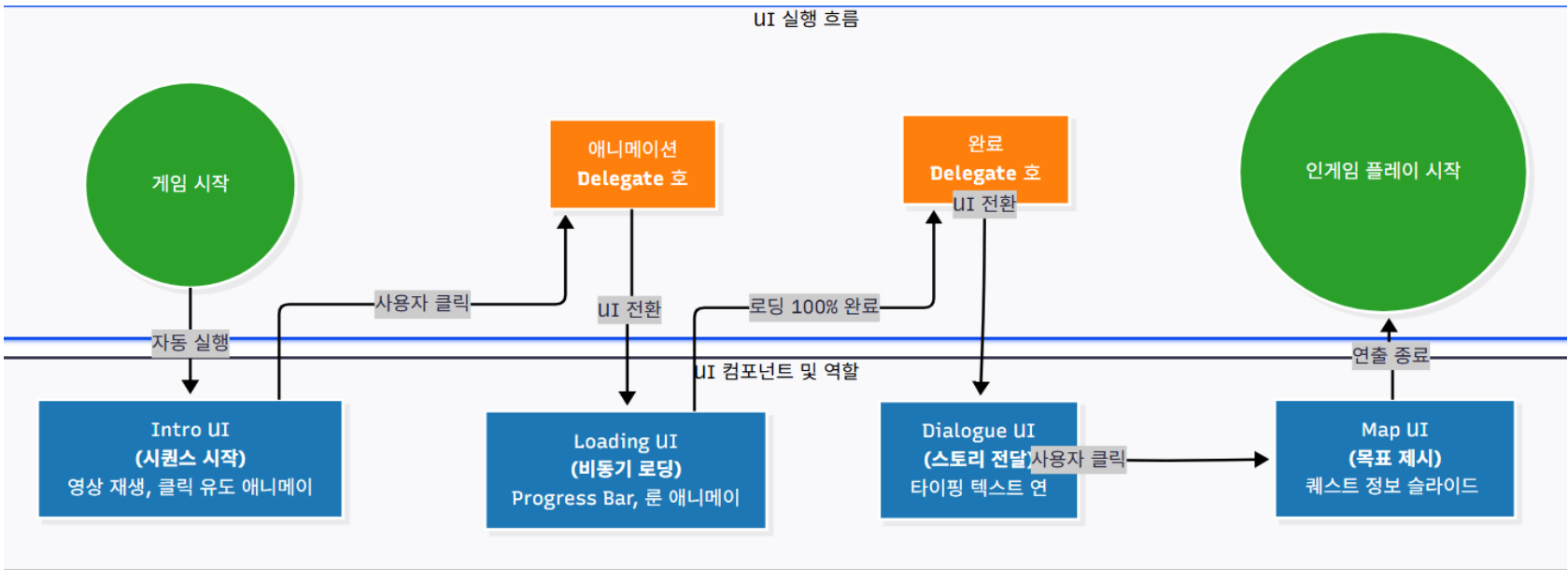
- 도입 배경
 1. 게임 시작부터 플레이어가 첫 퀘스트를 인지하기까지의 과정을 단절 없이 연결하여, 초반 몰입감을 극대화하는 시네마틱한 도입부 연출을 목표로 함.
- 구현 방식 및 역할
 1. 인트로 → 로딩: MediaPlayer로 영상 재생 후, 문구 Widget Animation 종료 시점에 FWidgetAnimationDynamicEvent Delegate를 바인딩하여 로딩 UI로 자동 전환.
 2. 로딩 → 대화: ProgressBar로 진행도를 표시하고, 로딩 완료 시 Delegate를 통해 대화창 UI를 활성화하여 끊김 없는 흐름을 유지.
 3. 대화 → 맵/퀘스트: 타이핑 효과로 텍스트를 출력한 뒤, 대화창 클릭 이벤트를 통해 맵 이름과 퀘스트 정보가 담긴 최종 UI를 출력하며 플레이어에게 제어권을 이관.
- 성과 및 효과
 1. 기능적으로 분리된 UI들을 유기적인 연출로 묶어냄으로써, 플레이어가 게임의 세계관과 스토리에 자연스럽게 몰입할 수 있는 환경을 제공.



옵션 UI



개발자 모드 UI



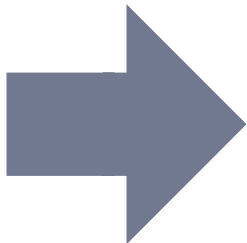
게임 시작 UI 흐름도

Trouble Shooting

프로젝트 트러블 발생 & 해결 과정

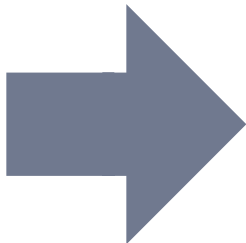
문제 발생 - 팀원 이탈

- 팀원 이탈로 인한 개발 범위 초과 문제
- 프로젝트 중간 팀원 이탈로 인해 기한 내 전체 기능 구현이 불가능해지고, 프로젝트 일정에 차질이 생길 위험 발생.



해결 과정

- 프로젝트의 핵심 재미를 기준으로 기능 우선순위를 재설정하고, 구현이 어렵지만 핵심이 아닌 기능은 덜어내는 '선택과 집중' 전략을 팀에 제안 및 설득.
- 기능별 중요도와 개발 리소스를 명확히 하여, 팀원 전원이 동의하는 현실적인 개발 로드맵 재수립.

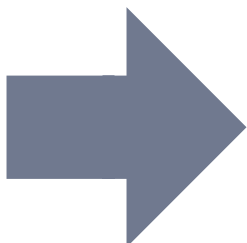


결과

- 핵심 기능에 개발 역량을 집중하여 메타버스 엑스포 출품 기간에 맞추어 사용자에게 설계된 플로우의 게임을 즐길 수 있게 됨

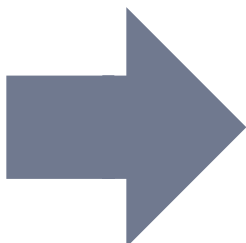
문제 발생 - 전투 시스템 피드백 부족

- 핵심 전투 시스템(패링/회피)의 낮은 타격감과 동기 부여 부족
- 패링, 저스트 회피 성공 시의 시청각적 피드백과 보상이 부족해 플레이어의 시스템 활용 동기가 부족했고, 전투가 단조롭게 느껴짐.



해결 과정

- 성공적인 리액션의 '쾌감' 극대화를 위해 레퍼런스(세키로)를 참고, 전용 VFX 및 SFX, Time Dilation을 활용한 '히트 스톱' 연출을 추가하여 타격감 강화.
- 성공적인 방어 후 강력한 '카운터 공격'으로 즉시 연계되도록 하여 '하이 리스크, 하이 리턴'의 재미 부여.



결과

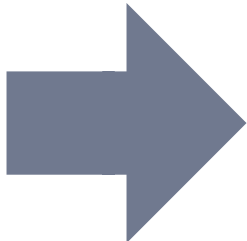
- 강화된 피드백과 명확한 보상으로 플레이어의 도전 욕구를 자극하고 전투의 깊이와 재미를 크게 향상.

Trouble Shooting

프로젝트 트러블 발생 & 해결 과정

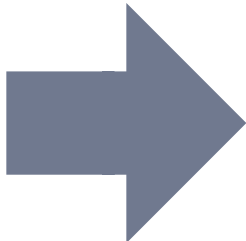
문제 발생 - 부정확한 피격 판정

- 빠른 모션으로 인해 피격 판정이 부정확해지는 문제 발생
- 단순 Collision 오버랩 방식으로는 빠른 공격 애니메이션과 캐릭터 움직임이 맞물릴 때 피격 판정이 누락되거나 부정확하게 발생하는 문제가 원인이었음



해결 과정

- Bézier Curve 기반의 궤적 추적 방식을 도입하여 판정 정밀도를 높임
- 검의 시작과 끝 Socket 위치를 기준으로, 프레임 사이의 이동 경로를 Bézier Curve로 보간, 정밀한 공격 궤적을 예측
- 예측된 궤적 공간을 여러 개의 삼각형 면으로 재구성하고, 해당 면 내부에 LineTraceByChannel을 실행하여 공간 기반의 정밀 충돌 판정을 구현

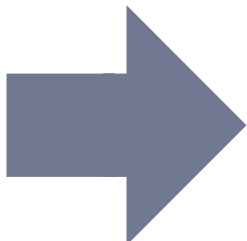


결과

- '맞지 않는 공격', '맞지 않아야 할 공격' 등의 부정적 경험을 제거하여 전투 몰입감 증대.

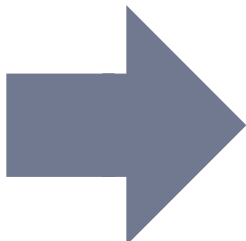
문제 발생 - 체감 난이도 상승

- '메타버스 엑스포' 시연에서 시연자들의 체감 난이도가 너무 높게 느껴지는 문제 발생.
- 시연의 특성상 발생한 문제. 예상 타겟(장르 숙련자)와 타 플레이어(장르 초심자)간의 장르 경험치 격차가 원인이었음.



해결 과정

- 튜토리얼 스테이지를 도입하고, 시연 현장에서 실시간 대응이 가능한 '**개발자 모드**'를 추가.
- 이후 시연을 거치며 얻은 플레이 데이터를 바탕으로 일자별로 개선, 난이도를 낮춘 'Easy Mode' 버전을 별도로 추가하여 접근성 확대.

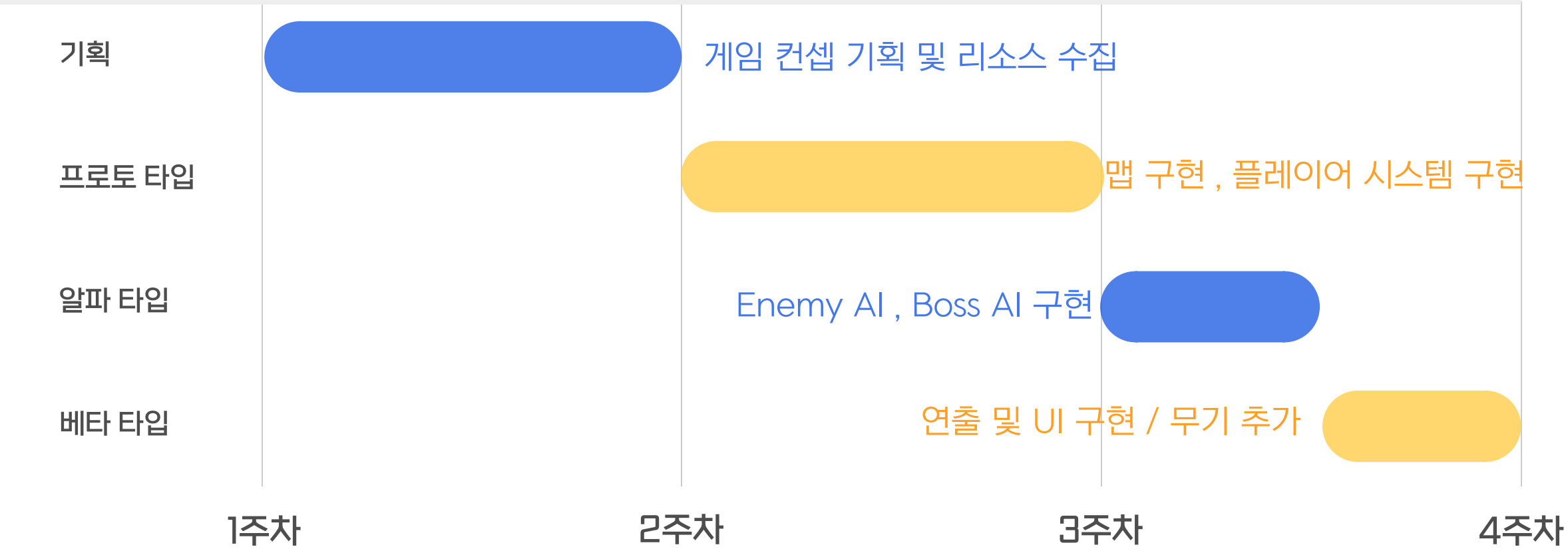


결과

- 시연 현장에서의 유연한 난이도 조절과 'Easy Mode' 제공을 통해, 초심자부터 숙련자까지 더 넓은 플레이어층이 긍정적인 게임 경험을 할 수 있도록 개선.

프로젝트 소개

장르	UNITY 기반 2D 탑다운 로그라이크 게임
기획 의도	단순한 연습용 게임이 아닌 상용 로그라이크 구조를 실제로 구축하는 것을 목표로 함.
게임 시스템	공격이 1타 2타로 분류되고 각 콤보 별로 다른 공격을 랜덤으로 선택하여 다양한 경험 제공
개발 기간	2023.11.23 ~ 2023.12.23
제작 인원	01명 (1인 개발)
사용 기술	UNITY / C# / GIT / GITHUB
GITHUB	깃 허브 이동하기



본인 구현 Gameplay Systems & Features

PLAYER 시스템	player.cs 기반 메인 컨트롤러, 8방향 이동 및 HP/HP Bar UI 연동 포탈 이동·상호작용·피격 처리(무적/카메라 흔들림)
무기 시스템	Weapon.cs로 공격·콤보·무기 방향 제어 WeaponSetting.cs로 12종 무기 데이터 및 스킬/데미지 보너스 관리
적 AI	EnemyController.cs로 배회→추적→공격 상태 전환 ScriptableObject 기반 EnemyData로 스탯·행동 관리
보스 시스템	BossController.cs로 3단계 패턴 AI·스폰·방어막 구현 특수 능력(스폰/배리어) 및 확률 기반 공격 분기
맵 생성	MakeRandomMap.cs로 프로시저럴 던전 생성 방 분할, 복도 연결, Tilemap 자동 배치
몬스터 스폰	Spawner.cs로 지역별 난이도 확장(몬스터 수·종류 증가) 모든 방 클리어 시 포탈 생성
UI/게임 관리	GameManager.cs로 전투/보스/메뉴 상태 제어 LevelUpMenuController.cs로 5종 스킬 강화 및 UI 연동
효과/사운드	코루틴 기반 화면 흔들림·투명도 변화, 파티클 전투 이펙트 배경/효과음 관리(AudioSource)



주요 구현 내용

■ 공격 패턴 다양화를 위한 다단계 무기 콤보 설계

- 도입 배경
 1. 단조로운 공격 방식을 피하고, 플레이어 선택에 따른 전투 변화를 구현하고자 함.
- 구현 방식 및 역할
 1. Weapon.cs: 공격 입력(좌Ctrl) → 2단계 연속 공격(1타/2타) 구현.
 2. WeaponSetting.cs: 무기별 프리셋(기본/스킬/특수 무기 총 12종).
 3. 플레이어 추적 무기 로직과 방향 전환 기능 제공.
 4. 같은 무기를 선택했을때는 공격력, 사거리, 공격속도 증가로 성장효과 부여
- 성과 및 효과

1타와 2타 조합에 따라 다른 플레이 경험을 제공하여 레벨업에 대한 보상 지급 및 레벨업 유도



레벨업 선택 UI

■ 확률 기반 다층 전투 패턴 보스 AI 설계

- 도입 배경
 1. 전투의 핵심 몰입도를 높이기 위해 연속 공격과 보스 전용 패턴 설계 필요.
- 구현 방식 및 역할
 1. BossController.cs: MoveTowardsPlayer()로 추적 AI.
 2. 공격 패턴 3단계 확률 기반 전환
 3. 배리어로 일시 무적 후 맵 기믹 수행 유도
 1. 기믹을 2분동안 진행하지 못한다면 전멸기 시전
- 성과 및 효과

전투 스타일 다양화로 플레이어 전략성 강화

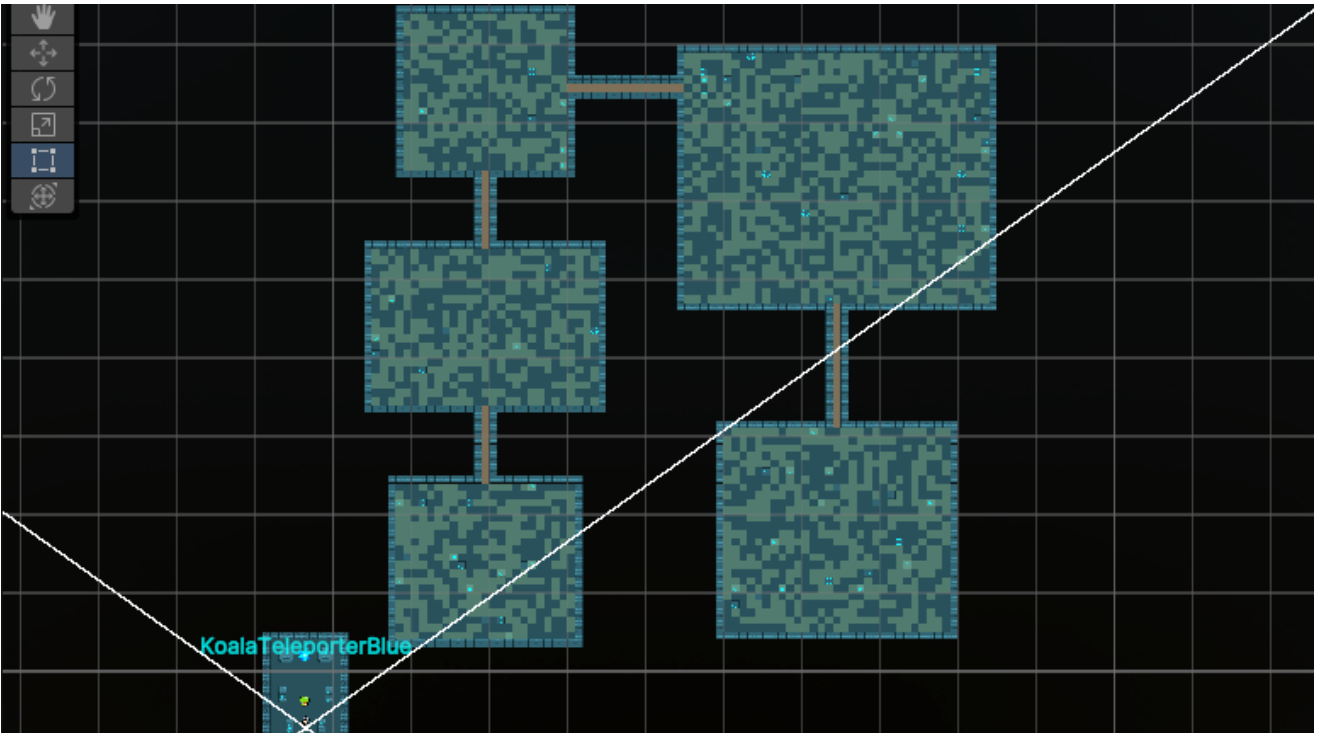


보스전 인게임

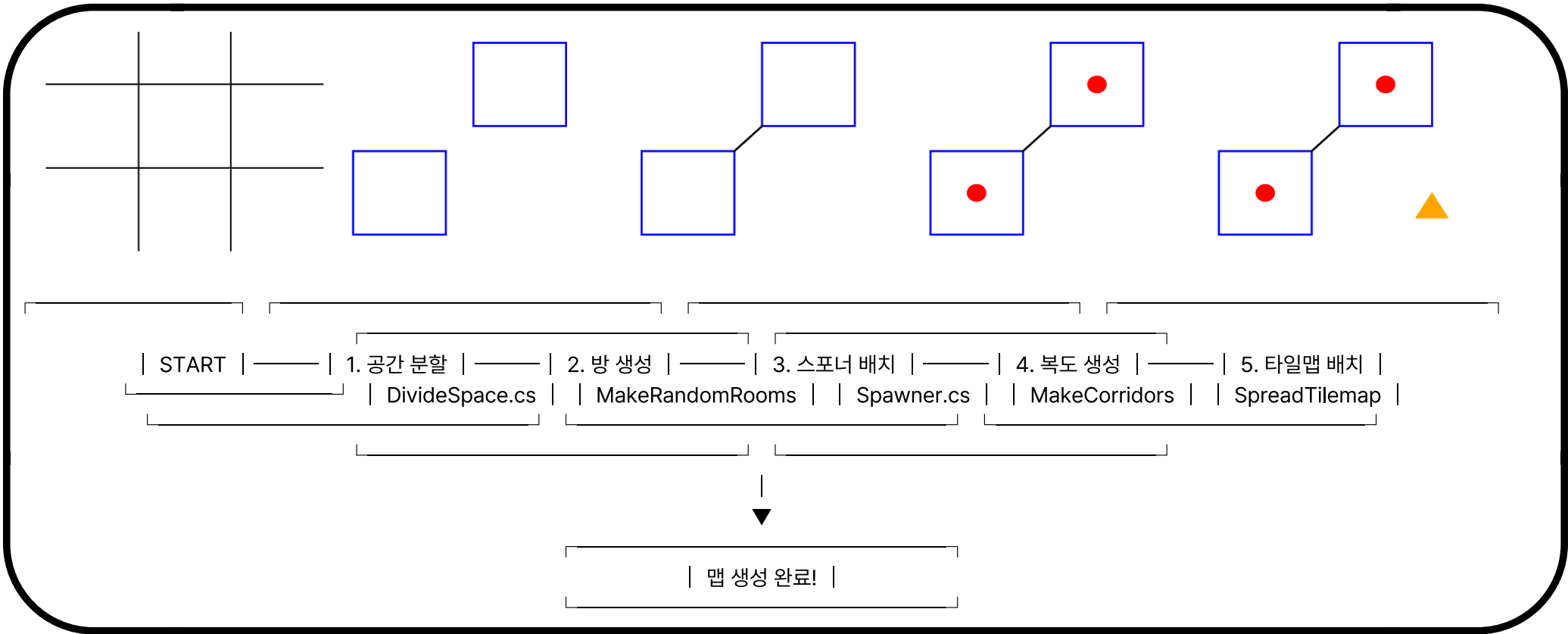
주요 구현 내용

■ 비선형 탐험성을 위한 랜덤 공간 분할 기반 던전 생성 기법

- 도입 배경
 1. 로그라이크 특성상 매 판마다 다른 던전 구조 필요.
 2. 단순 반복 맵에서 오는 피로도를 줄이고, 탐험 동기 부여 강화.
- 구현 방식 및 역할
 1. DivideSpace.cs : 맵 전체를 재귀적으로 분할하여 최소 단위 방 영역 생성
 2. MakeRandomRooms.cs : 랜덤 크기의 방을 배치, 방의 중심 좌표 계산
 3. Spawner.cs : 각 방에 몬스터 스포너 배치, 난이도에 따른 적 수 조절
 4. MakeCorridors.cs : 방과 방을 연결하는 복도 생성, 자동 타일 배치
 5. SpreadTilemap.cs : 바닥·벽·복도 타일 배치, 던전 전체 구조 완성
 6. MakePortals.cs : 포탈 배치로 마을 ↔ 던전 이동 가능
- 성과 및 효과
 - ✓ 매 판마다 다른 던전 구조 → 새로운 탐험 경험 제공
 - ✓ 특정 테마(예: 불/얼음) 추가 규칙으로 손쉬운 확장 가능
 - ✓ 랜덤 던전 구조 자동화 → 리플레이 가치 상승



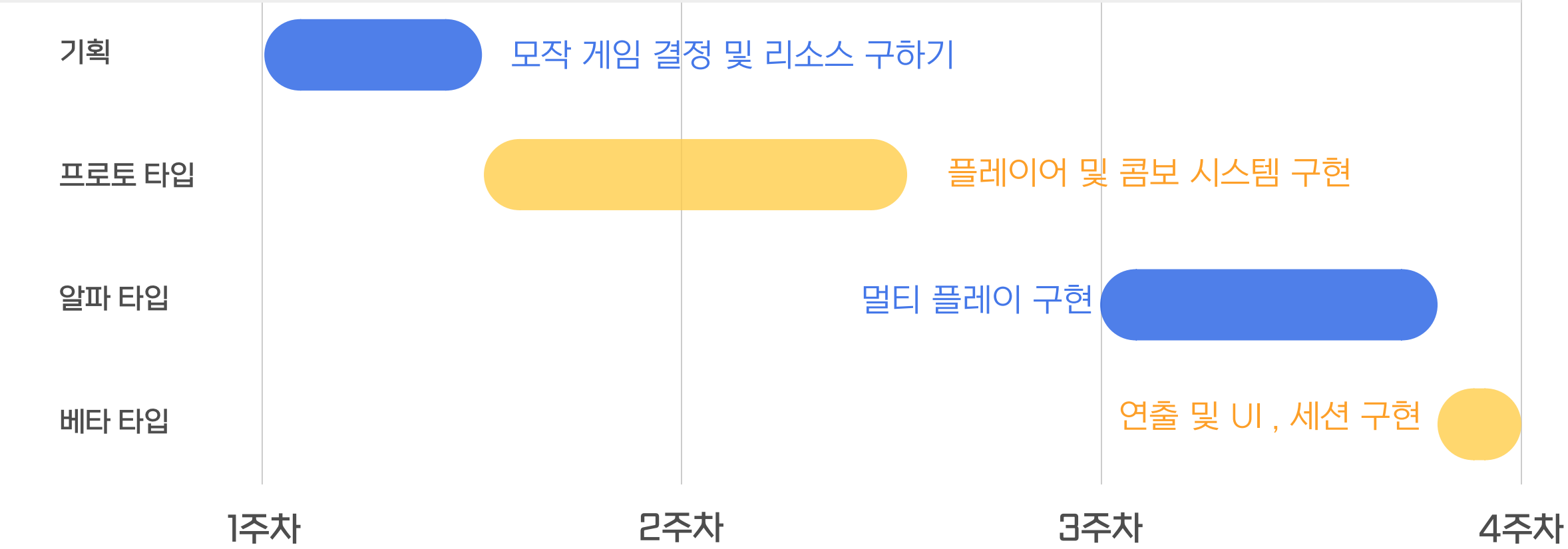
랜덤으로 생성된 맵



맵 생성 순서도

프로젝트 소개

장르	UNREAL ENGINE 기반 헌팅 액션 RPG
기획 의도	4인 멀티로 진행되는 몬스터 헌터 월드의 무페토 지바전 모작
구현 목표	멀티 플레이와 대검의 참모아베기 구현 목표
개발 기간	2024. 03. 01 - 04. 01
제작 인원	02명 (클라이언트 개발자 2명)
사용 기술	UNREAL ENGINE5 / C++ / GIT / GITHUB
GITHUB	깃 허브 이동하기



본인 구현 Gameplay Systems & Features

PLAYER 시스템	Hunter.cpp로 플레이어 컨트롤러/HP/카메라/피격 처리 구현 무기 교체(ChangeWeapon), 피해 처리(TakeDamage), 네트워크 동기화(RPC)
무기 시스템	WeaponComponent.cpp로 Dash/Quick/Heavy/Unique/Roll 입력 처리 WeaponDataAsset으로 무기 데이터 관리, RPC로 공격 동기화 GreatSword.cpp 전용 콤보/노티파이 공격 판정
멀티플레이어 시스템	Steam/로컬 네트워크 기반 4인 협동 지원 세션 생성/검색/참가, 호스트-클라이언트 이동 및 상태 관리
UI 시스템	QuestBoardWidget: 방 생성/참가 UI, 동적 세션 목록 생성 HunterMainWidget: 체력바·게임 상태 표시, SessionButtonWidget 연동
네트워킹	클라이언트-서버 호스트 방식, RPC 함수 복제 위치·애니메이션·공격 상태 실시간 동기화
전투 시스템	Quick/Heavy/Unique 3타입 공격, 애니 노티파이 기반 콤보 Roll 회피·무기 교체·정밀 타이밍 판정 구현
효과/사운드	코루틴 기반 화면 흔들림·투명도 변화, 파티클 전투 이펙트 배경/효과음 관리(AudioSource)



주요 구현 내용

멀티플레이어 네트워킹 기반의 협동 전투 시스템

- 도입 배경
 1. 실시간 협동 플레이: 최대 4명의 플레이어가 동시에 Safiijiiva 보스와 전투할 수 있는 협동 시스템 구현
 2. 플레이어 위치, 공격 상태, 애니메이션을 실시간으로 동기화하여 일관된 게임플레이 보장
- 구현 방식 및 역할
 1. 세션 관리 시스템: MHGameInstance에서 Steam/로컬 네트워크 기반 세션 자동 생성, 방 검색 및 참가, 자동 서버 이동
 2. 실시간 동기화: RPC 기반 통신, 상태 복제, CharacterMovementComponent를 활용한 부드러운 플레이어 이동 동기화
 3. 협동 전투 시스템: 공유 보스 체력, 팀 데미지, 플레이어 간 역할 분담과 타이밍 조율을 통한 전략적 전투
- 성과 및 효과
 - ✓ 사용자 경험: QuestBoardWidget을 통한 간편한 방 생성/참가, 실시간 피드백, 로비에서 게임으로의 자연스러운 전환
 - ✓ Steam 네트워킹을 통한 안정적인 연결, 4명 동시 플레이어 지원으로 확장성 확보



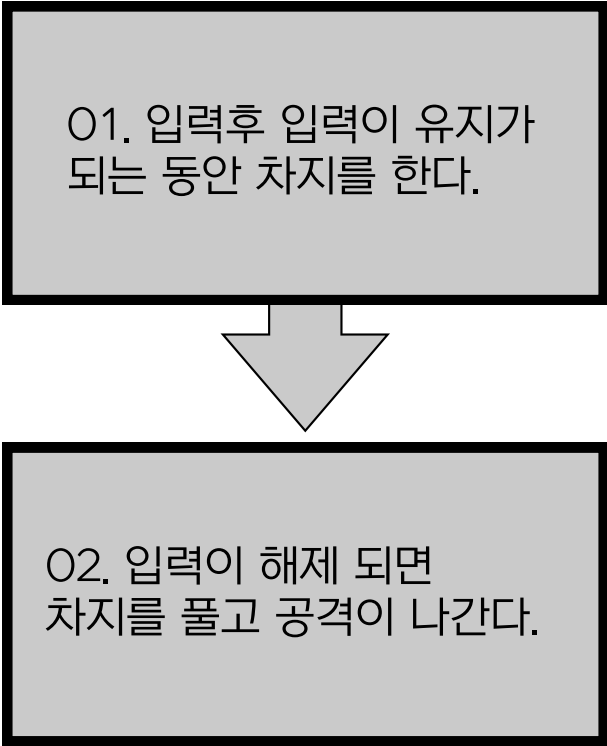
멀티플레이 플레이 캡처



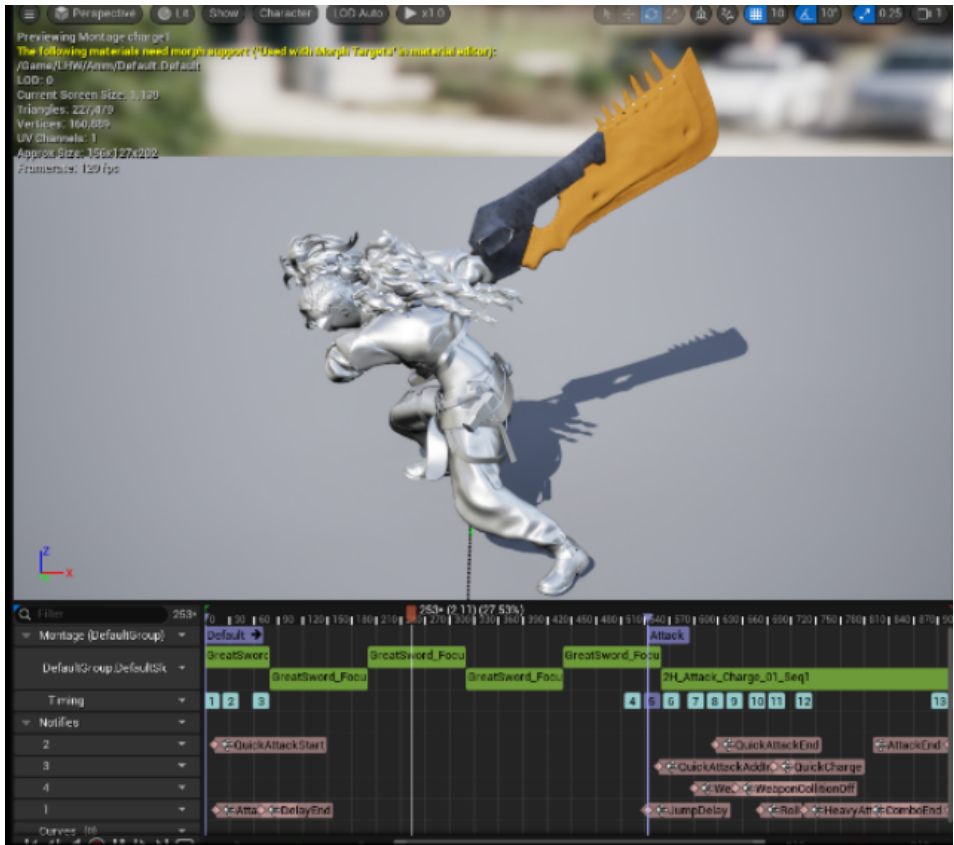
세션 UI 페이지

주요 구현 내용

■ 콤보 시스템(참 모아베기) 구현

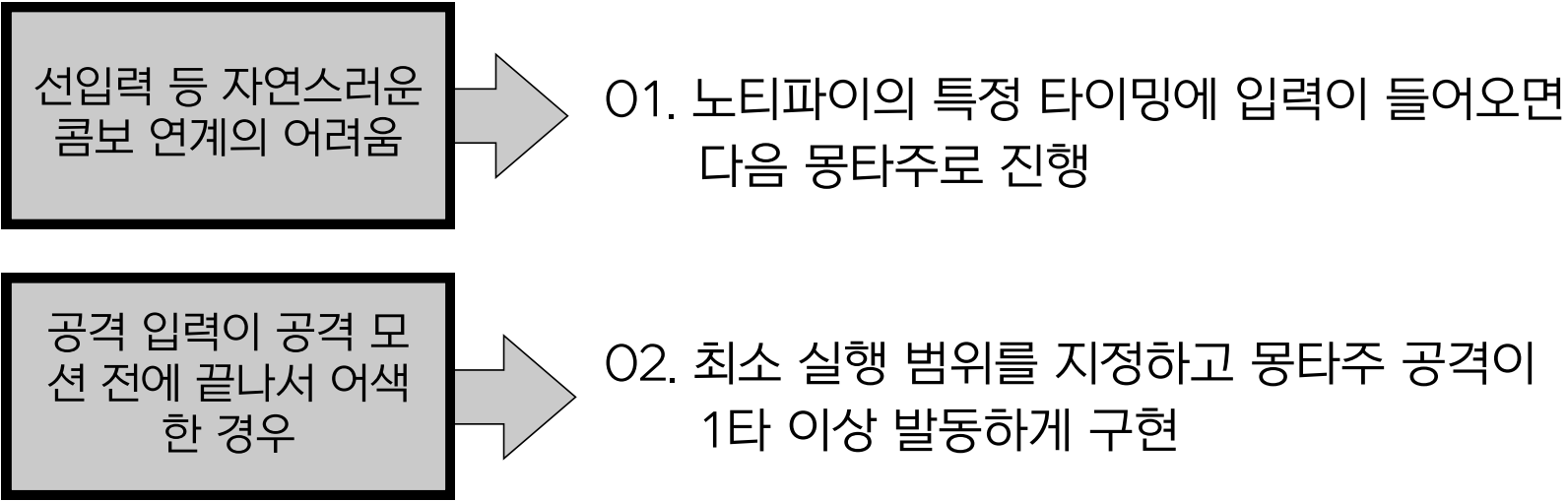


모아베기, 참모아베기 구현 목표



모아베기, 참모아베기 구현 목표

■ 콤보 시스템의 문제발생 상황

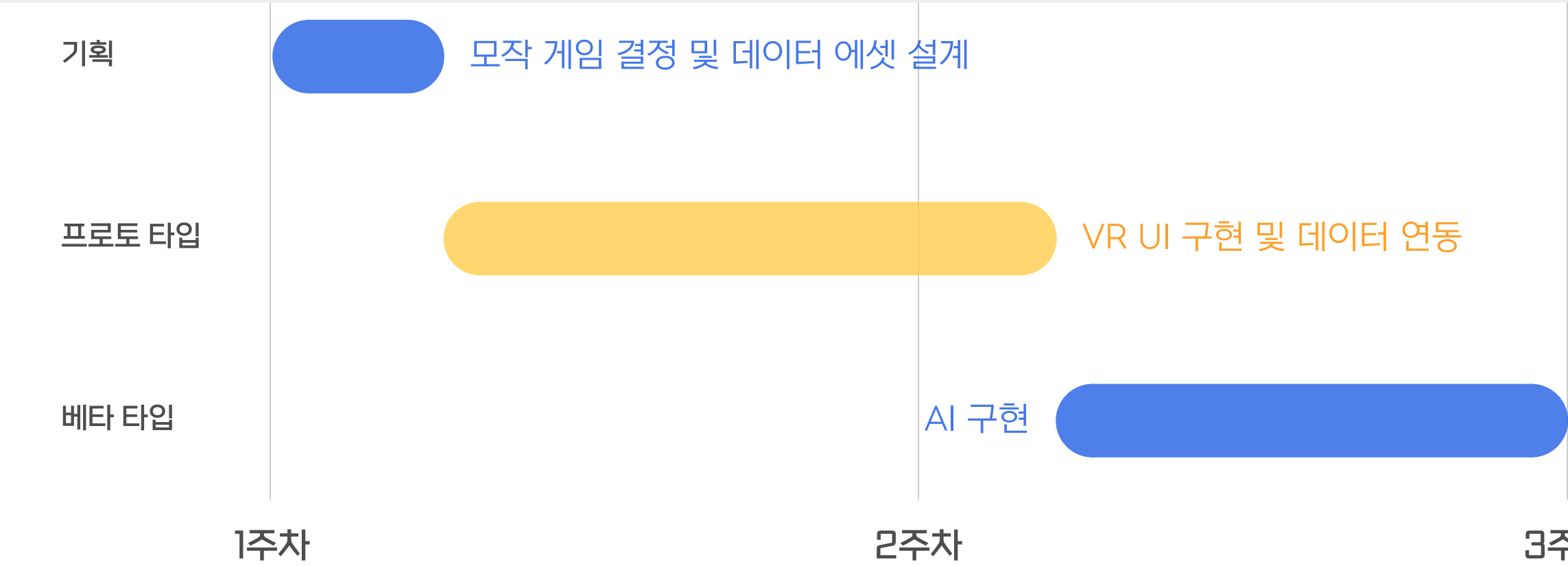


■ 최종 개선 목표

- 01. 노티파이 스테이트를 사용 하였으면 노티파이 및 코드 복잡성을 완화
- 02. 공격시작, 차지, 공격 상태를 만들어 관리하여 가독성을 개편
- 03. 델리게이트를 사용해 입력시 비동기 처리하여 더욱 간단하게 구현 예정
- 03 - 1. 앞에서 서술한 프로젝트 오로라 제작 과정에 개선, 영향을 줌

프로젝트 소개

장르	UNREAL ENGINE 기반 VR 시뮬레이션 게임
기획 의도	1인칭 PC 플레이 기반인 게임을 VR로 이식하여 몰입감을 증대
구현 목표	SuperMaket Together VR 모작
개발 기간	2025. 03. 18 - 2025. 04. 01 (3주)
제작 인원	02명 (클라이언트 개발자 2명)
사용 기술	UNREAL ENGINE5 / C++ / GIT / GITHUB
GITHUB	깃 허브 이동하기

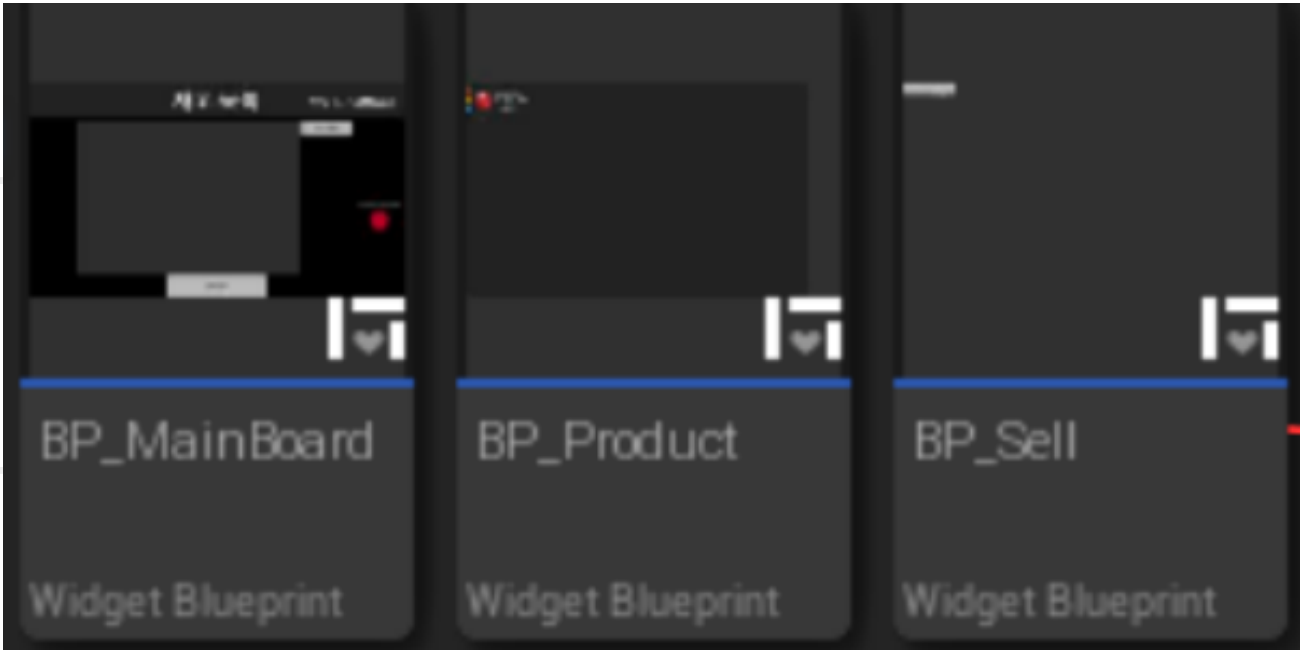


본인 구현 Gameplay Systems & Features

AI 시스템	Behavior Tree 기반 AI 의사결정 시스템 BlackboardComponent로 상품 탐색·이동 로직 관리 AI Task Node(이동, 상품 선택, 계산대 이동) 모듈화
AI 캐릭터 행동	상품 진열대 상호작용으로 상품 획득 계산대 도착 시 데이터 전달 및 대기열 관리 Overlap 이벤트로 자동 상호작용 처리
상점 관리	상품 수량별 동적 메시 변경 및 태그 관리 데이터 자산 기반 상품 정보(재고·가격·메시) 일원화
상품 박스 시스템	ProductBoxSpawner/Actor로 박스 자동 생성 구매 시 상품 박스 자동 배치 및 물리 처리
UI/위젯 시스템	메인 보드 위젯: 상품 정보·구매·매출/재고 관리 계산대 모니터 위젯: 키패드 입력·결제 검증 상품 정보 위젯: 상품 상세·선택 처리
시스템 아키텍처	EProductType/EAIState 등 열거형 기반 데이터 설계 컴포넌트 기반 구조(충돌 감지, 메시, UI 표시) GameMode와 연동된 전역 상태·대기열·매출 관리
게임플레이 특징	AI 고객의 자동 상품 탐색·구매 및 결제 진열대 재고 자동 관리·상품 자동 배치 실시간 재고·매출·AI 행동 모니터링



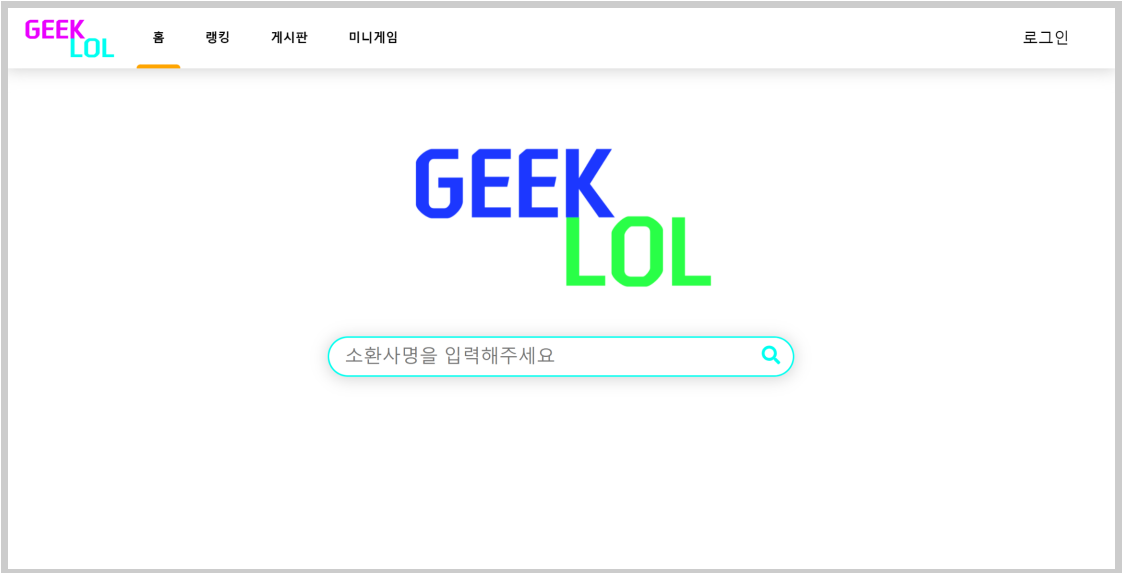
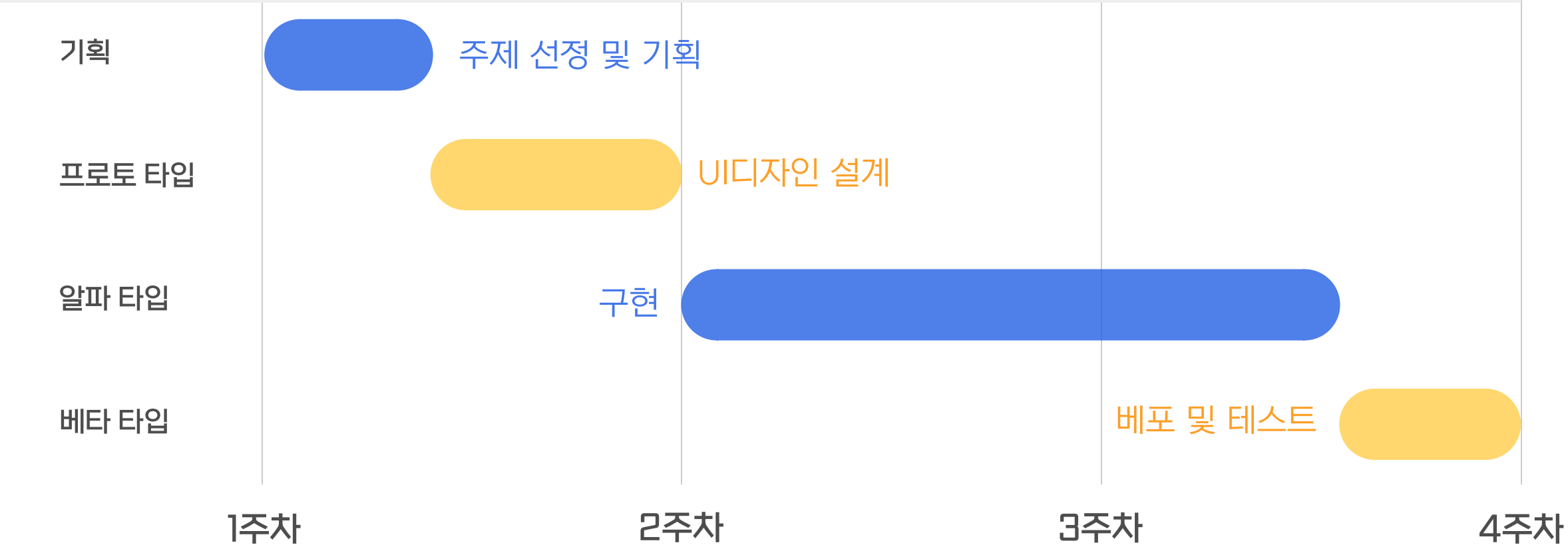
상점 관리 UI



UI 모듈

프로젝트 소개

페이지 용도	롤 전적검색 & 커뮤니티 사이트
기획 의도	기존 OP.GG에서 LCK 팀별 자유게시판과, SHORT, 롤문철 페이지 추가
구현 목표	OP.GG 모작
개발 기간	2024.01.18 ~ 2024.02.18
제작 인원	6인 제작 (프론트 2 백엔드 2 풀스택 2)
사용 기술	JAVASCRIPT / REACT / SPRING / JAVA / HTML / AWS
GITHUB	<u>깃 허브 이동하기</u>

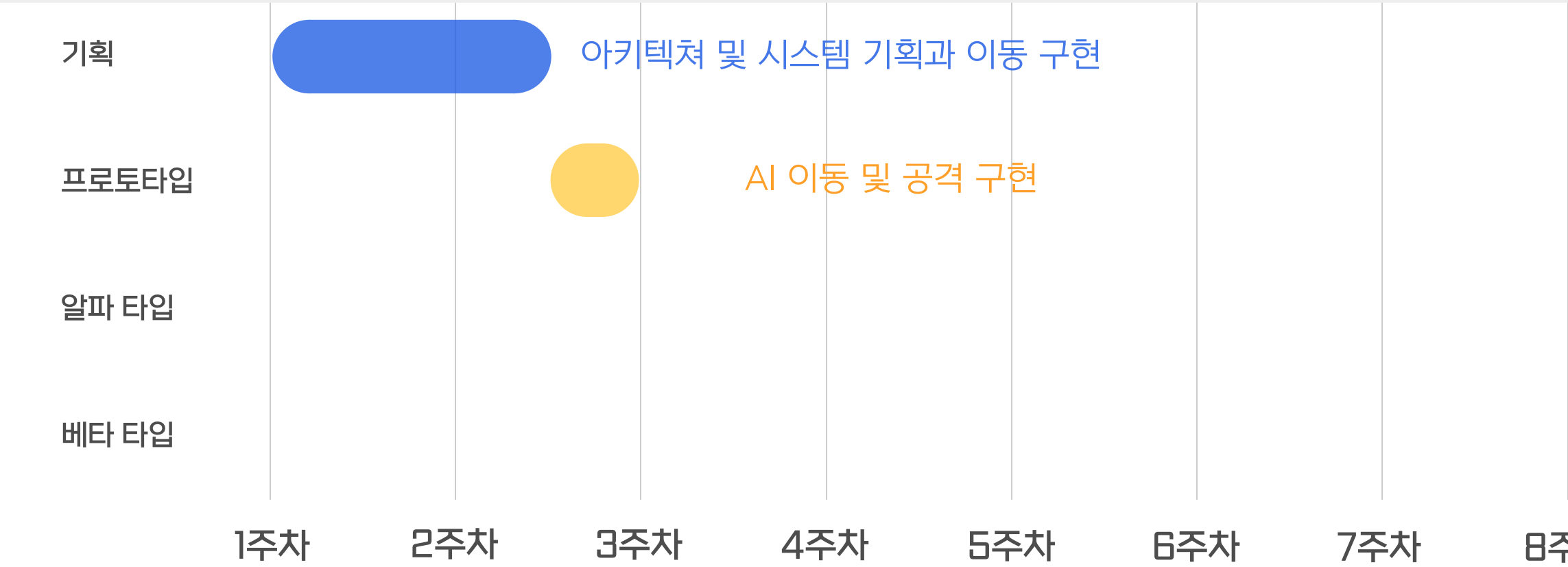


본인 구현 Gameplay Systems & Features

컴포넌트 아키텍처	모든 UI 기능을 React Component로 분리 ,단일 책임 원칙 기반 설계로 재사용성과 확장성 확보
라우팅 시스템	React Router 기반 페이지 전환/렌더링, useNavigate, useLocation 훅으로 동적 라우팅 구현
상태 관리	useState로 로컬 상태 제어, useEffect로 데이터 페칭/사이드 이펙트 처리, 상태 변경 시 UI 자동 업데이트
API 통신	fetch API 활용 RESTful 서버 연동, JWT 기반 인증·권한 관리, 안정적 에러 핸들링
인증/권한 관리	JWT 인증 및 자동 로그인, localStorage 세션 관리, Google OAuth 연동, RBAC(Role Based Access Control) 적용
게시판 시스템	자유게시판, LCK, 솔루션, 트롤신청 게시판 통합 관리, 검색·정렬·페이징 기능, 재사용 가능한 게시판 컴포넌트 구조
마이페이지 시스템	프로필 관리, 계정 수정, 활동 내역 조회 기능 분리, 사이드 메뉴 기반 내비게이션, 사용자 활동 통계 시각화
검색/필터링	다중 조건 검색(제목/내용/작성자), 정렬 옵션(최신순/인기순), 실시간 결과 업데이트 및 페이징 처리
파일 업로드	FormData API로 멀티파트 업로드, 프로필 이미지 변경, 썸네일 미리보기 기능 구현
에러/피드백	HTTP 상태 코드 기반 에러 처리, 입력 유효성 검사·실시간 피드백·로딩 상태 관리
UI/UX 디자인	SCSS 기반 모듈화 스타일링, Material-UI와 react-icons으로 일관성 있고 직관적인 UI 구성

프로젝트 소개

장르	액션 슈팅 소울라이크 (PVE, TPS)
기획 의도	정교한 전투 메커니즘과 직관적인 슈팅 조작이 결합된 액션 게임 구현 및 출시
구현 목표	State Tree를 이용한 보스 AI 시스템 구현 목표
개발 기간	2025. 08. 01 ~ 2025. 09. 27
제작 인원	04명 (클라이언트 개발자 3명, 기획자 1명)
사용 기술	UNREAL ENGINE5 / C++ / GIT / GITHUB
GITHUB	깃 허브 이동하기



읽어주셔서
감사합니다

THANK YOU FOR CHECKING OUT
MY PORTFOLIO.

LEE HYOWON

GAMECLENT, LEE HYO WON

010 - 2327 - 9714

reppy0813@gmail.com