PORTFOLIO

안녕하세요,
'될 때까지'가 아닌, '*될 수밖에 없게*' 만드는 기획자,
안현선입니다.

Name: 안현선
Phone: 010-9386-6439
Contact: kiwibird0417@gmail.com

Profile

안현선 2000.04.17

010-9386-6439

kiwibird0417@gmail.com

https://github.com/kiwibird0417

기술 스택

프로그래밍 언어	C++, C#
엔진	언리얼 5.5, 유니티6, 유니티 2022
협업 툴	Git, Figma, Google Sheets, JIRA, Slack, Discord

학력사항

2019.03~2024.02	숙명여자대학교 일본학과 숙명여자대학교 IT공학전공(복수전공)
2016.03~2019.02	서문여자고등학교 졸업

교육 이력

2024.12~2025.06	언리얼 엔진과 생성형 AI를 활용한 게임 개발자 양성 과정 / 청년취업사관학교 새싹
2024.08~2024.12	유니티 공인 자격증까지 1+1, XR메타버스 창작자 실전 취업캠프 과정 / 청년취업사관학교 새싹

행사 참여 및 자격증, 수상 이력

2025.06	메타버스 엑스포 3일간 부스 참가 - Textoria
2024.12	UNITY ASSOCIATE 자격증
2022.07	1치킨 알만툴 게임잼(2022) - 그래픽상 3등

Projects

Unreal Engine 5.5

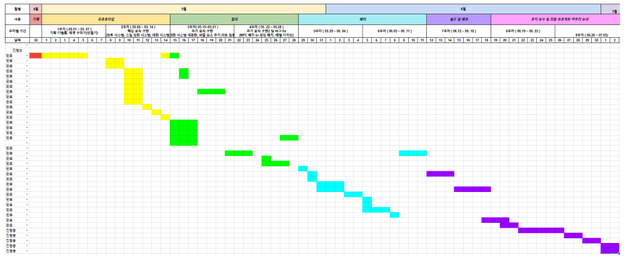
01. Textoria ····· 메트로바니아 플랫포머 게임 - UE 5.5 (p4)
02. Mabinogi Mobile(PNM) ····· 마비노기 모바일 모작 - UE 5.5 (p4)
03. Subnautica ····· 서브나우티카 VR 모작 - UE 5.5 (p11)
04. War of Gods ····· 갓오브워 토르 보스전 모작 - UE 5.5 (p15)
05. StartRail! ····· 붕괴 스타레일 전투 시스템 - UE 5.5 (p21)

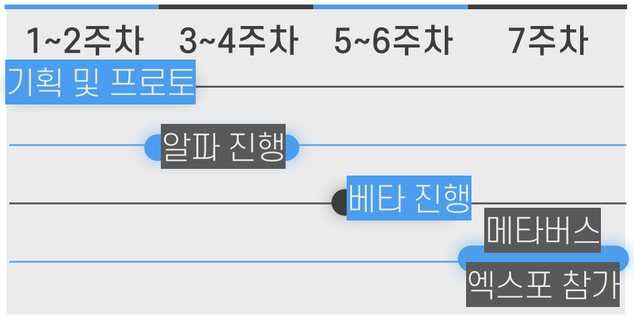
Unity 6

06. Scan the Beat! ····· 바코드 스캐너 VR 리듬 게임 - Unity 6 (p24)

01. Textoria - 메트로바니아 플랫포머 게임

https://youtu.be/UWs2nSHJOSk





진행 기간 2025.05.01~2025.06.27(7주)

소개 창작 게임 메트로바니아 플랫포머 게임 Textoria 제작

개발 규모 3인

담당 파트

기획 : 화면설계서 제작, 월드 레벨 디자인, 플로우 차트 작성

개발 : 대화 시스템 구현, 월드 상호작용 구현, 저장 시스템 구현(Game Instance)

아트 : 플레이어 및 에너미 스프라이트 제작, 월드 및 보스전투 음악 구현

구현 환경

엔진: Unreal Engine 5.5

개발환경: C++

관리: Github, Google Sheets

기타: ClipStudio, Figma

01. Textoria - 기획 의도

https://www.canva.com/design/DAGotBTXgWE/gJR0X0vOJFdpNJhM0Hltnw/edit?utm_content=DAGotBTXgWE&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

- 감성 중심의 메트로바니아 플랫포머 구성

  이야기를 써 내려가듯 구성된 스토리 기반의 2.5D 액션 어드벤처 RPG

  → 플레이어는 세계를 구하기 위한 여정을 '문장사'라는 설정 속에서 진행

  → 기존 메트로바니아 장르의 어두운 분위기에서 벗어나, 따뜻한 분위기로 차별화

- 고양이와 문장사의 독특한 조합

  샴 고양이와 문장사를 결합한 독창적인 세계관

  → 샴 고양이를 주인공으로 내세워 기존 캐릭터들과의 차별화 시도

  → '문장사'라는 신선한 소재를 통해 플레이어의 호기심을 자극하고 몰입도 높임

- 밝고 따뜻한 색감의 비주얼 아이덴티티

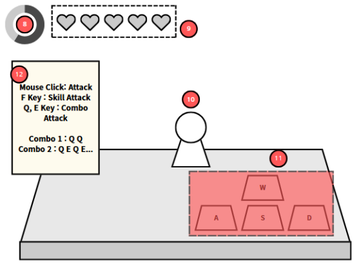
  기존 장르 대비 시각적 차별화

  → 어두운 던전 중심의 기존 메트로바니아와 달리, 밝고 감성적인 색채를 적극적으로 활용

01. Textoria - 화면 설계와 플로우 차트

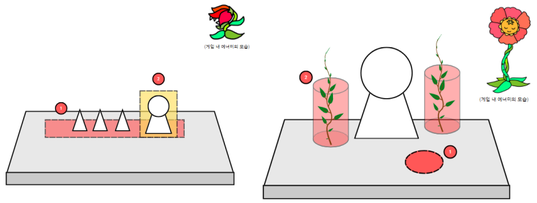
https://www.canva.com/design/DAGrVHqmfvQ/eHXaa6kgHRzqFzC-WEoxBQ/edit?utm_content=DAGrVHqmfvQ&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

[플레이어 메인 화면 UI 설계]

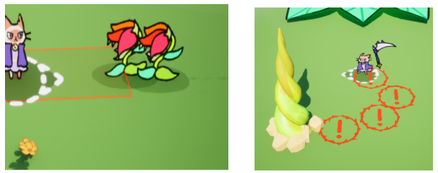


[실제 화면]

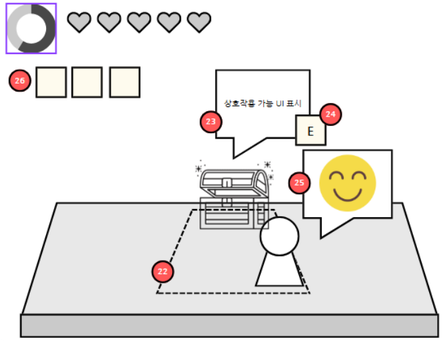
[에너미 패턴 설계]



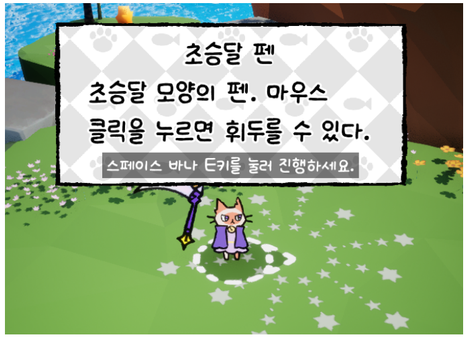
[실제 화면]



[상호작용 화면 설계]



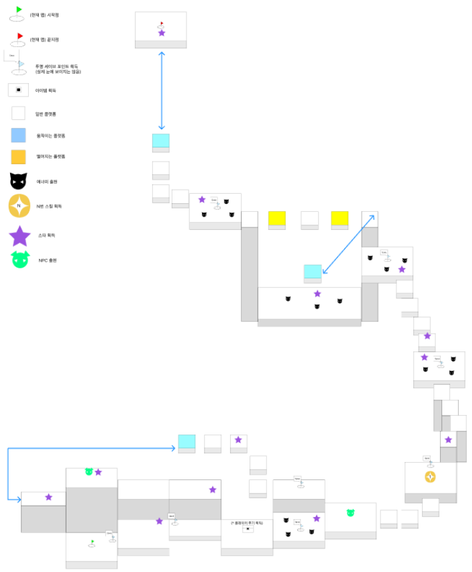
[실제 화면]



- 플레이어 메인 화면 및 스킬 화면 UI 설계
  - 표시되어야 할 플레이어 체력, 마나, 별 개수 표시 설계 및 구현
- 일반 및 보스 에너미 패턴 설계
  - 화면에서 보스 에너미가 공격하는 방식 및 표현 방식 설계
- 아이템 및 대화 상호작용 구현
  - Game Instance에 정보값을 기록하여, 중복으로 무기와 스킬을 획득하는 것을 막음
  - 대화 상호작용의 경우, DialogueComponent를 만들어서, 캐릭터별 ID와 Quest ID를 부여하여, 데이터 테이블에서 읽어와 출력되는 방식으로 구현함
  - 시연 시 피드백을 바탕으로 현재 말하는 캐릭터에게 카메라 포커스 및 스킵 기능 추가

01. Textoria - 레벨 디자인 및 설계

https://www.canva.com/design/DAGotBTXgWE/gJR0X0vOJFdpNJhM0Hltnw/edit?utm_content=DAGotBTXgWE&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton



- 일반 맵 및 보스 맵 레벨 디자인 구현
  - 기획 단계에서 수립한 맵 내 기믹 배치(떨어지는 플랫폼, 움직이는 플랫폼 등)

- 세이브 포인트 구현
  - 플랫폼별로 물로 떨어지게 되면, 이전에 가장 가까이 있던 Savepoint Actor를 기록
  - 리스폰 시 저장된 Savepoint Actor가 있다면 해당 위치에서 스폰, 아니라면 초기 위치에서 리스폰

Map ID	Map Location
1	/Game/Maps/Menu_Levels/TitleMap_001.TitleMap_001
2	/Game/Maps/Menu_Levels/Opening_Sequence_002.Opening_Sequence_002
3	/Game/Maps/Menu_Levels/Ending_Sequence_003.Ending_Sequence_003
101	/Game/Maps/Spring_Levels/Spring_Water_Map_101.Spring_Water_Map_101
102	/Game/Maps/Spring_Levels/Spring_Water_Map_102.Spring_Water_Map_102
103	/Game/Maps/Spring_Levels/Spring_Water_Map_103.Spring_Water_Map_103
500	/Game/Assets/LUSH_StylizedEnvironmentSet/Foliage/Flowers/Spring_Sky_Island_Map_500.Spring_Sky_Island_Map_500
503	/Game/Maps/Spring_Levels/Sky_Islands/Boss_Sprint_Sky_Island_Map_503.Boss_Sprint_Sky_Island_Map_503
501	/Game/Maps/Spring_Levels/Sky_Islands/Spring_Sky_Island_501.Spring_Sky_Island_501
502	/Game/Maps/Spring_Levels/Sky_Islands/Spring_Boss_Map_502.Spring_Boss_Map_502

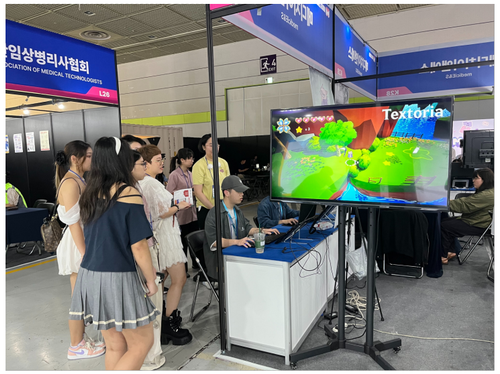
- 맵 ID를 활용한 워프 구현
  - Game Instance에 MapID와 실제 레벨 위치를 저장한 데이터 테이블 활용
  - 해당 맵 이름에서 가장 오른쪽 "_"를 기준으로 MapID를 판단, 해당 ID의 맵으로 이동하는 함수를 어디서든 부를 수 있도록 설정

01. Textoria - 메타버스 엑스포 참가

https://docs.google.com/spreadsheets/d/1JV2tb-h3fRBT58CsIzZ89csQlkVrI6lIPP0Dk_BQsfg/edit?usp=sharing

- 메타버스 엑스포 3일 참가(6월 18일~6월 20일)
  - 총196명 체험, 91명의 설문조사

- 종합적 평가 요약
  - 강점: 감성적인 그래픽, 귀여운 고양이 캐릭터, 독특한 배경 콘셉트에 긍정적 반응
  - 개선점: 액션 시스템의 조작감, 슈퍼 점프 및 스킬 사용의 직관성, 보스전 타격감 보완 필요



- **참여 소감**

1. 유저와의 직접 소통을 통한 몰입감 있는 피드백 경험

단순히 게임을 만드는 것에서 그치지 않고, 유저가 어떻게 받아들이는지를 직접 듣는 경험은 매우 소중했음

→ 평소 추상적으로만 상상했던 유저 반응을 실제 피드백을 통해 구체화할 수 있었음

→ "내가 귀엽다고 생각했던 캐릭터를 유저도 귀엽다고 느꼈다"는 공감의 순간이 큰 동기부여가 되었음

2. 부족한 점을 마주한 순간, 성장의 방향이 생김

액션 조작의 불편함, 직관성 부족, 전투 리듬 등에 대한 지적은 뼈아프면서도 매우 명확한 개선 방향 제시였음

→ 피드백 덕분에 '어떤 점을 수정하면 플레이 경험이 좋아질까'를 구체적으로 계획할 수 있었음

→ 단순히 완성도를 높이는 게 아닌, *진짜 플레이어 중심의 설계를 고민하게 된 계기*

3. 완성된 결과보다 '공유된 경험'이 더 강하게 남았음

개발자로서 '보여주기 위한 무대'가 아니라, 같이 웃고 같이 놀아주는 공간을 만들었다는 자부심

→ "내가 만든 세상을 누군가가 좋아해주었다"는 사실은, *수치 이상의 값진 성취였음*

01. Textoria - 결과 및 회고

https://youtu.be/UWs2nSHJOSk

1. 개발 과정에서의 어려움과 해결 방법

- 저장 정보 관리:

  - Game Instance를 활용하여 아이템 획득, 대화 진행, 이벤트 트리거 등 주요 진행 데이터를 관리하는 시스템 구성

  - 데이터 테이블과의 연동, 씬 전환 간 데이터 유지, 리스폰 시점의 상태 복구 등의 로직을 Game Instance에서 처리하도록 구현

  →플레이어가 죽었을 때, 레벨을 다시 로드하고, 이전 상태를 자연스럽게 이어가도록 구조 설계에 집중하여 시스템 안정성 확보

- Paper ZD와 애니메이션 방향 처리 :

  - 2D 캐릭터의 방향에 따라 다양한 공격 애니메이션을 적용하는 과정에서 PaperZD의 구조와 연동 방식 이해에 시간 소요

  - 애니메이션의 방향 판별 로직, 조건 노드 설정, 방향별 전환 타이밍 등을 수동으로 조정

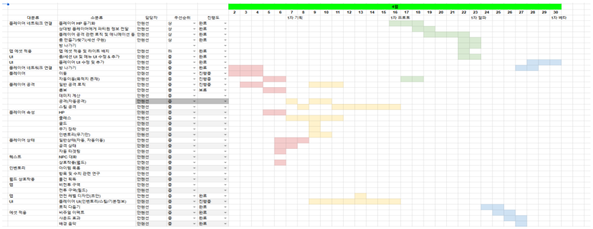
2. 피드백 및 개선 방향

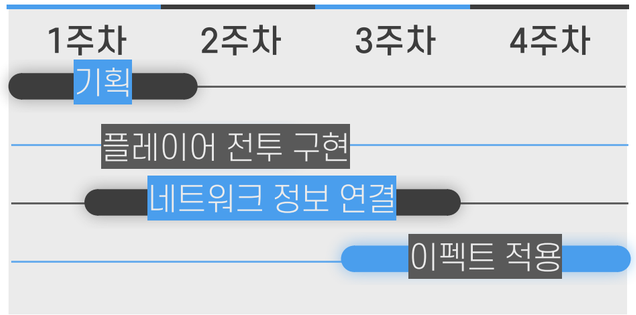
- 메타버스 엑스포 참여 후 시연자 피드백:

  - 현장에서 플레이하는 시연자들의 피드백을 통해 난이도 조정, 추가로 언어 지원 및 저장 기능을 추가로 구현하여 더 자연스러운 게임플레이 제공

02. Mabinogi Mobile(PNM)

https://github.com/kiwibird0417/PNM_KFC_MabinogiMobile
https://cafe.naver.com/ictacademy/13111





진행 기간	2025.04.02~2025.04.30(4주)
소개	마비노기 모바일 던전 전투 모작
개발 규모	2인
담당 파트	기획 : 플레이어 캐릭터 및 전투 설계, 던전 레벨 디자인 개발 : 플레이어 캐릭터 이동 및 상호작용, 전투 로직 구현
구현 환경	엔진: Unreal Engine 5.5 개발환경: C++ 관리: Github, Google Sheets 기타: ClipStudio, Figma

02. Mabinogi Mobile(PNM) - 기획 의도

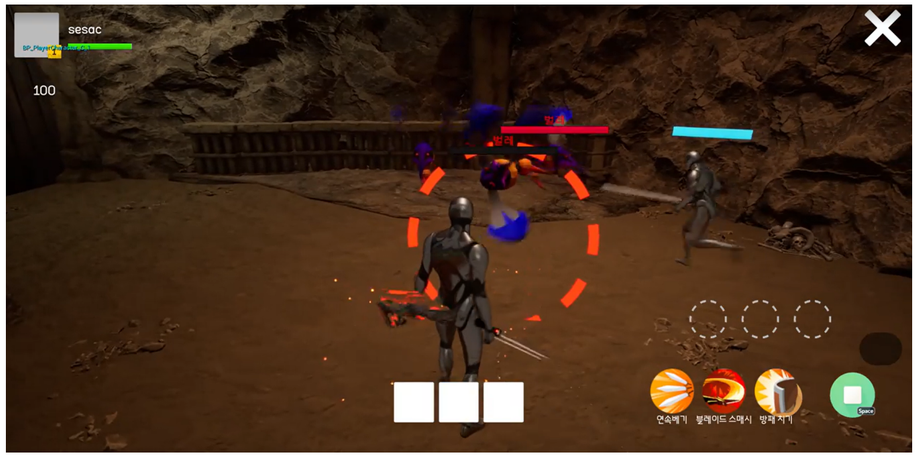
https://github.com/kiwibird0417/PNM_KFC_MabinogiMobile
https://cafe.naver.com/ictacademy/13111

- 전략적 전투 시스템:

  - 직업별 특화 스킬과 전략적 조합

  → 각 직업의 특징을 살린 스킬 시스템을 통해 플레이어는 다양한 조합을 통해 전투의 흐름을 익히며 재미를 경험

  → 단순한 공격/방어가 아닌, 직업 간 시너지 효과와 전략적 사고를 요구하는 전투 환경을 제공



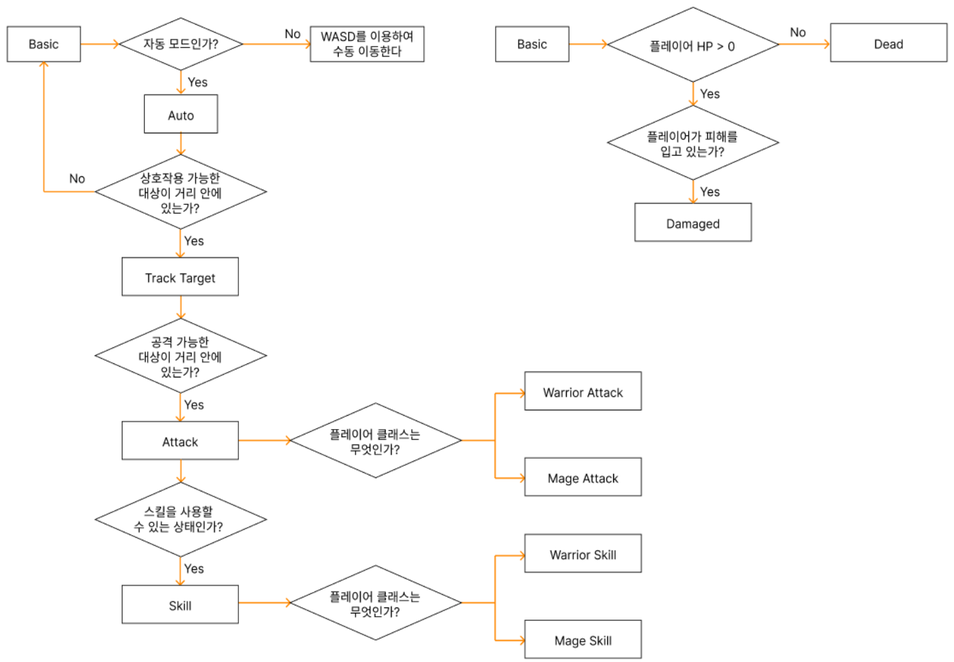
- 실시간 협동 전투 시스템:

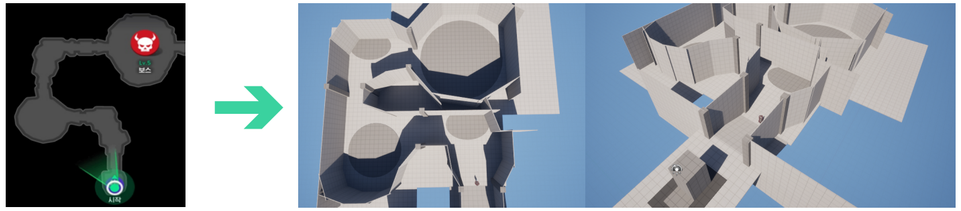
  - Unreal Engine에서 제공하는 Play as Listen Server 방식을 이용하여 다양한 전투 전략을 요구하는 멀티플레이

  → 실시간 협동 전투 시스템을 통해 팀원 간의 협력이 중요한 요소로 작용 던전 내 보스전 및 이벤트 전투에서 각 플레이어의 역할 분담이 중요하며, 이를 통해 협동 플레이 강화

02. Mabinogi Mobile(PNM) - 게임 설계

https://github.com/kiwibird0417/PNM_KFC_MabinogiMobile
https://cafe.naver.com/ictacademy/13111





- 플레이어 구현
  - Actor Component를 활용한 플레이어에서 실행하는 기능 관리
  - 플레이어 이동 시 입력, 클래스(직업), 스킬, 인벤토리를 구분하여 체계적으로 시스템을 구분하여 관리

- 플레이어 상태 구현(FSM)
  - 원작의 자동 전투 및 이동을 플레이어의 행동을 State로 구분하여 구현
  - 상호작용 가능한 목표물이 존재하면, 거리에 따라 자동으로 록온 상태가 되어 추적하게 되고, 에너미가 공격 가능한 범위만큼 가까워지면 공격을 실행
  - 플레이어는 공격 상태에서 스킬 버튼으르 수동적으로 사용하여 전투에 참여

- 맵 구성 및 레벨 디자인
  - 원작의 알비 던전 1층 1번방의 레벨 디자인을 참고한 레벨 디자인 설계

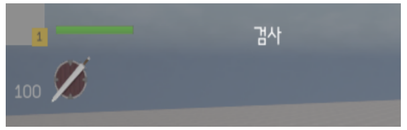
02. Mabinogi Mobile(PNM) - 게임 설계

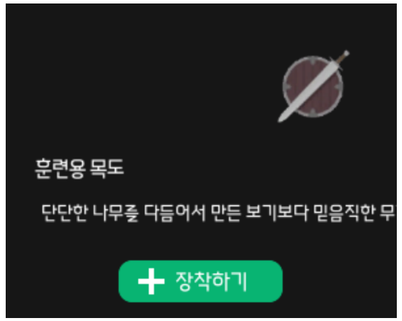
https://github.com/kiwibird0417/PNM_KFC_MabinogiMobile
https://cafe.naver.com/ictacademy/13111

불러온 클래스 데이터 정보에 따라
제대로 스킬 아이콘이 보임

- 플레이어 스킬 및 데이터 테이블
  - csv 파일을 언리얼의 데이터 테이블과 연동시켜 체계적으로 관리
  - 실제 전투 시스템에서 사용되는 스킬 쿨타임, 아이콘, 데미지 계수를 미리 데이터 테이블에 기록하여 계산할 때 용이하도록 설계

  → 공통적으로 활용할 수 있는 Skill Interface를 생성하여, 클래스별 스킬 구현 시 같은 형식에서 다른 클래스의 스킬들이 UI에 반영되도록 설계





플레이어는 인벤토리에서 무기를 선택하면,
클래스가 변경되고, 장착된 무기가 UI 및 실제로 모델에 장착됨

- 플레이어 무기 장착 및 인벤토리 구현:
  - 플레이어 클래스별 무기와 클래스에 따라 다른 애니메이션 연결 및 로직 구현
  - MMORPG를 목표로 하여 대표적인 클래스로 전사와 마법사를 선정하여 근거리와 원거리 공격 로직을 원작처럼 자동 공격으로 구현
  - 장착 무기는 인벤토리에서 관리하고, 데이터 테이블로 유연성 있게 대응하여 새로운 무기 추가 및 스킬 계수 등에 영향을 미치는 값 수정 가능

02. Mabinogi Mobile(PNM) - 결과 및 회고

https://github.com/kiwibird0417/PNM_KFC_MabinogiMobile
https://cafe.naver.com/ictacademy/13111

1. 개발 과정에서의 어려움과 해결 방법

- 전투 밸런스 조정:
  - 던전 전투 시스템에서 각 직업의 스킬과 능력치 밸런스 맞추기에 어려움
  - 플레이 테스트를 여러 번 진행하며, 다양한 직업 간 조합과 역할 분담을 고려하여 밸런스를 조정함

  → 실시간 피드백을 통해 각 직업의 역할과 스킬이 던전 내에서 균형을 맞추도록 최적화

- 협동 전투 시스템 동기화:
  - 실시간 협동 전투에서 여러 명의 플레이어 간 동기화를 맞추기 어려웠음
  - 플레이어 간 행동에 대한 실시간 피드백을 강화하여 동기화 문제 최소화

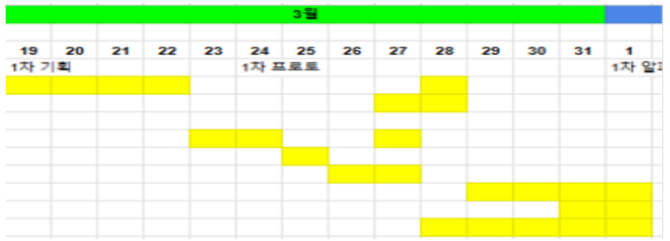
2. 피드백 및 개선 방향

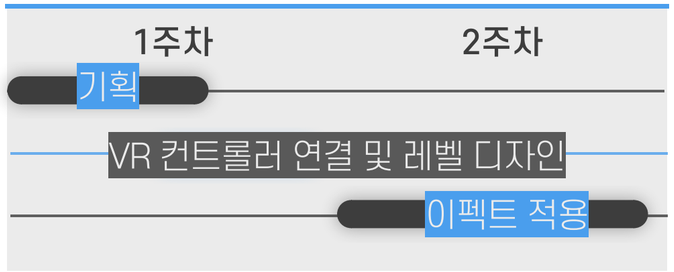
- 직업별 스킬의 다양성 및 직업 간 시너지 강화:
  - 직업 간 시너지 효과에 대한 피드백을 반영하여, 각 직업의 스킬이 서로 보완할 수 있도록 개선.

- 전투 환경의 몰입감 향상
  - 던전의 환경과 전투 효과에 대한 피드백을 통해, 더욱 몰입감 있는 전투 환경을 구현할 수 있도록 개선. 던전에 전투 효과를 추가하여 시청각적인 몰입감을 높임

03. Subnautica

https://github.com/kiwibird0417/SubnauticaVR
https://cafe.naver.com/ictacademy/13071





진행 기간 2025.03.19~2025.04.01(2주)

소개 서브나우티카 VR 모작

개발 규모 2인

담당 파트

기획 : 월드 설계

개발 : 플레이어 캐릭터, 에너미 및 물고기 AI, UI 제작 및 애니메이션, 카메라 세팅, 에셋 적용

구현 환경

엔진: 언리얼

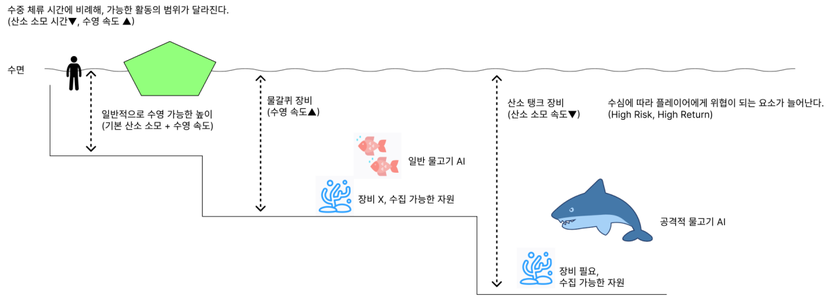
관리: GITHUB, GOOGLE SHEETS

개발환경: C++, 블루프린트

03. Subnautica - 기획 의도

https://github.com/kiwibird0417/SubnauticaVR
https://cafe.naver.com/ictacademy/13071

- 탐험 및 생존 요소 구현:
  - 자원 채집 및 제작 시스템 구현 → 심해 생존의 핵심 루프 재현
  - 기지 건설 기능 추가 → 플레이어가 자율적으로 거점을 마련하고 확장하는 경험 제공

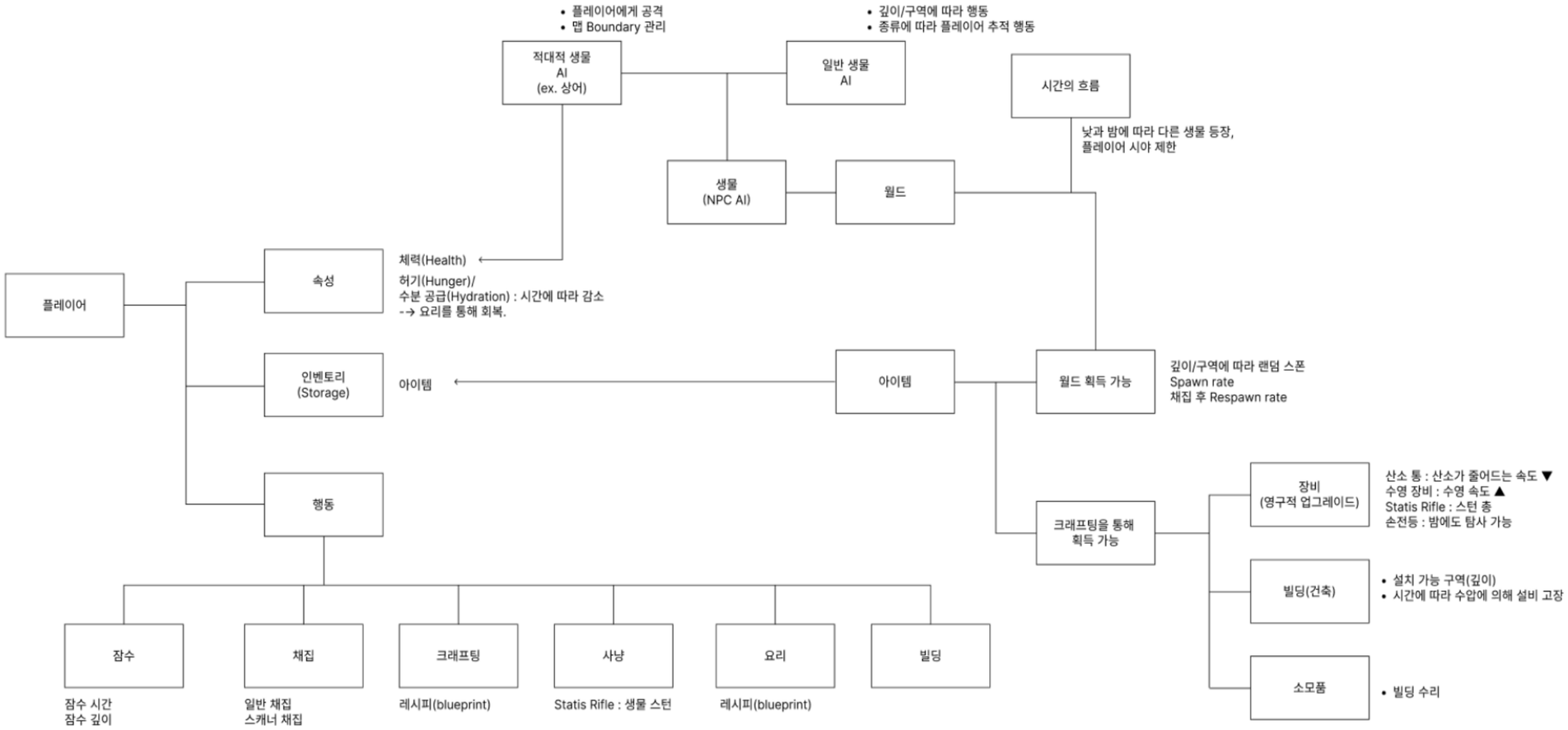


- 심해 생태계 연출:
  - 다양한 해양 생물 AI 구현 → 플레이어와의 상호작용 설계
  - 심해의 공포감을 강조하는 연출 기법 적용 (시야 제한, 음향 디자인 활용)

- VR 환경 최적화:
  - 1인칭 시점의 몰입도를 강화하기 위해 핸드 트래킹 및 인터랙션 시스템 구현
  - 심해 특유의 공간감을 살리기 위한 렌더링 및 조명 최적화

03. Subnautica - 구조 분석

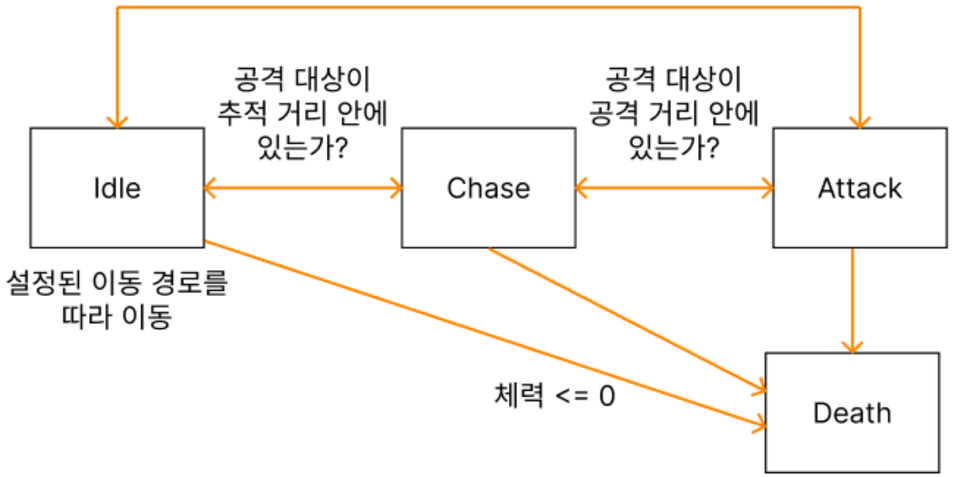
https://github.com/kiwibird0417/SubnauticaVR
https://cafe.naver.com/ictacademy/13071



03. Subnautica - 게임 설계

https://github.com/kiwibird0417/SubnauticaVR
https://cafe.naver.com/ictacademy/13071

	Row	ID	Name	Type	Area	Rarity	Required Equipment	Crafting Materials	Effects	Description
1	101	101	Stone	Resource	Upper	Common	None			A basic resource.
2	102	102	Metal Scrap	Resource	Upper	Common	None			Crafting material.
3	103	103	Copper	Resource	Middle	Rare	Pickaxe			Needed for crafting electronic components.
4	104	104	Silver	Resource	Middle	Rare	Pickaxe			Needed for crafting advanced equipment.
5	105	105	Crystal Fragment	Resource	Lower	Very Rare	Laser Cutter			Used for upgrading advanced equipment.
6	106	106	Battery	Resource	Middle	Common	None			Used for powering electronic devices or equipment.
7	201	201	Knife	Tool		Common		Metal Scrap x2	Attack Power +10	Basic melee weapon. Can attack enemies up close.
8	202	202	Fins	Tool		Common		Rubber x2, Cloth x1	Swim Speed +20%	Increases swimming speed. Helps to move efficiently underwater.
9	203	203	Oxygen Tank	Tool		Common		Metal Scrap x3, Silicone x1	Oxygen +50%	Increases underwater time by boosting oxygen levels.
10	204	204	Laser Cutter	Tool		Rare		Crystal Fragment x1, Battery x1	Enables specific resource gathering	A tool for gathering specific resources and minerals.
11	205	205	Deep Sea Oxygen Tank	Tool		Very Rare		Silver x2, Silicone x2	Oxygen +100%	Essential for deep-sea exploration, greatly increases oxygen supply
12	206	206	Flashlight	Tool		Common		Battery x2	Light +20%, Expands View	Provides light in dark areas and expands vision.
13	207	207	Pickaxe	Tool		Common		Metal Scrap x2		



- 아이템 데이터 테이블
  - CSV 파일을 언리얼의 데이터 테이블과 연동시켜 체계적으로 관리
  - 항목은 ID, 이름, 종류, 획득 구역, 희귀도, 필요장비, 제작 재료, 효과, 설명으로 구성

- 에너미 AI
  - 기본적으로 SPLINE 이동 경로를 따라 순찰
  - 플레이어가 감지 범위 내 접근하면 플레이어를 추격하여 공격하며,
    감지 범위에서 벗어나면 다시 이동 경로로 복귀

- 맵 구성 및 환경 디자인
  - POST PROCESS VOLUME 및 FOG를 활용한 해저 공간 구현
  - PCG 기능 및 FOLIAGE를 활용한 맵 구성

03. Subnautica - 결과 및 회고

https://github.com/kiwibird0417/SubnauticaVR
https://cafe.naver.com/ictacademy/13071

1. 개발 과정에서의 어려움과 해결 방법

- 수영 속도 구현의 난관:
  - 심해 환경에서 현실적인 수영 속도를 구현에 어려움을 겪음
  - 물속 저항값을 계산하는 방식이 자연스럽지 않아서, 최종적으로 에디터에서 조정 가능한 값으로 설정하여 해결

- VR 멀미 문제:
  - 포스트 프로세스 볼륨을 활용해 더 몰입감 있는 환경을 만들려 했으나, VR에서 멀미가 심하게 발생하여 결국 제외
  - VR 환경에서는 시각적 효과보다 이동 방식과 UI 디자인이 멀미 최소화에 더 중요한 요소임을 체감

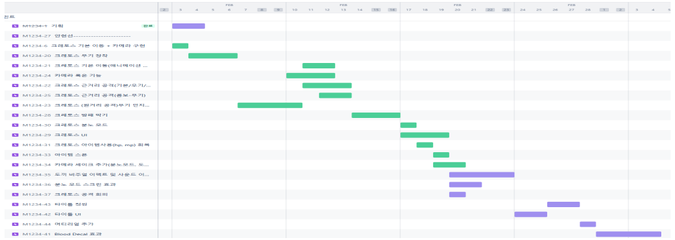
2. 피드백 및 개선 방향

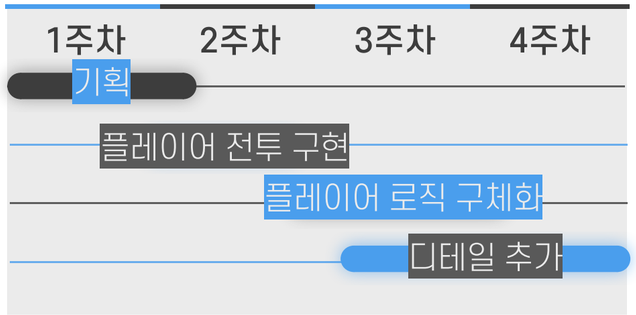
- 데이터 테이블 확장성 문제:
  - 현재 CSV 파일을 통해 아이템 데이터를 미리 불러오는 방식을 사용
  - 라이브 서비스 환경에서는 시즌별 업데이트가 존재하므로 이를 고려한 확장성이 필요

  → DB 연동 및 동적인 데이터 로딩 구조 도입 검토

04. War of Gods

https://github.com/kiwibird0417/WarofGods
https://cafe.naver.com/ictacademy/13034





진행 기간 2025.02.03~2025.03.06(5주)

소개 갓오브워 라그라노크 토르 보스 전투 모작

개발 규모 2인

담당 파트

기획 : 플레이어 캐릭터 기능 설계 및 구현

개발 : UI 제작 및 애니메이션, 카메라 세팅, 에셋 적용

구현 환경

엔진: 언리얼

관리: GITHUB, JIRA

개발환경: C++, 블루프린트

04. War of Gods - 기획 의도

https://github.com/kiwibird0417/WarofGods
https://cafe.naver.com/ictacademy/13034

- 전투 시스템 구현:

  - 근거리 및 원거리 전투 시스템 설계 → 크레토스의 타격감과 속도 반영

  - 공격, 회피, 방어, 무기 전환 자연스럽게 연결 → 유기적인 전투 흐름 구현

- 조작감 및 몰입도 개선:

  - 전투 및 애니메이션 속도, UI 개선 → 플레이어가 직관적으로 정보를 파악하도록 설계

  - 연출 최적화 → 몰입감을 극대화하는 전투 애니메이션 및 이펙트 적용

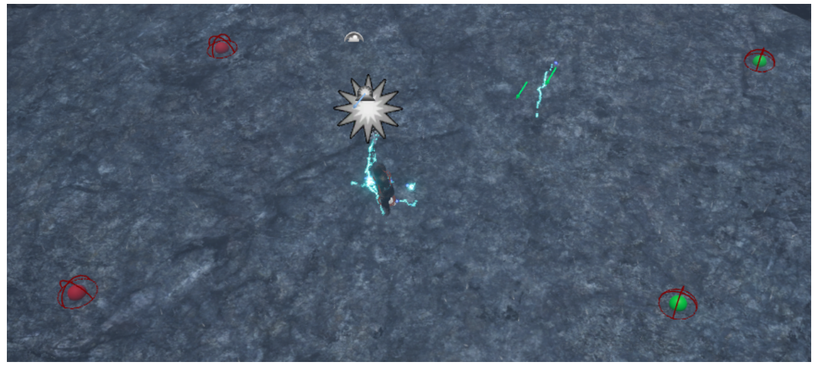
- 실시간 액션 전투 연구:

  - 원작의 조작감 및 전투 디자인 분석 → 핵심 요소를 체험하며 연구

  - 반응성과 타격감의 중요성 탐구 → 실시간 전투에서의 손맛과 피드백 강화

04. War of Gods - 아이템 드랍 설계

https://github.com/kiwibird0417/WarofGods
https://cafe.naver.com/ictacademy/13034



- 전투 아이템 등장 확률 조정 시스템

  - HP 회복 아이템 등장 확률 : 플레이어 HP가 낮을 수록 회복 아이템 등장 확률 증가

  - 분노 게이지 회복 아이템 등장 확률 : HP가 충분할 경우, 분노 게이지가 부족하면 분노 회복 아이템 등장 확률 증가

  → 플레이어 상태 기반 아이템 배치, 전투 지속성 및 전략성 강화

- UI 및 상호작용 개선

  - 아이템 사용 시 가시성 문제 해결 → "PRESS E" UI 추가하여 직관적인 상호작용 제공

04. War of Gods - 결과 및 회고

https://github.com/kiwibird0417/WarofGods
https://cafe.naver.com/ictacademy/13034

1. 개발 과정에서의 어려움과 해결 방법

- 플레이어 속도 및 타격감 조정:
  - 전투 몰입도를 유지하기 위한 조작감에 어려움을 겪음
  - 세부적인 애니메이션 속도를 조정하여 플레이어와 에너미 간의 리듬감을 구현하려고 노력

- 플레이어와 에너미 간의 상호작용:
  - 플레이어가 공격을 하지 못하는 피격, 에너미가 피격을 하는 모션이 연속되자 어색함을 느낌
  - 플레이어 및 에너미의 공격뿐만 아니라 각 액터의 피격 처리까지 고려하여 전투를 구성해야함을 깨닫고 고려하여 구현

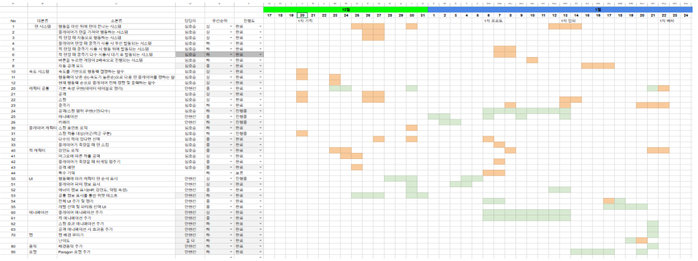
2. 피드백 및 개선 방향

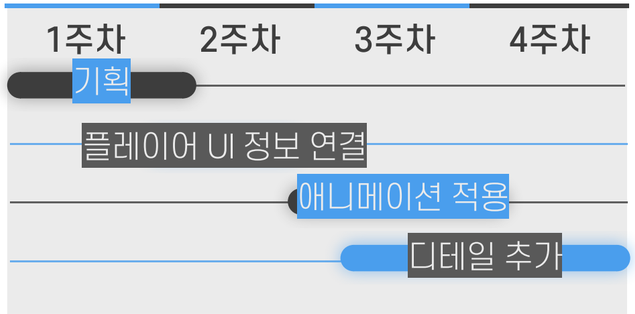
- 테스트 플레이를 통한 개선
  - 팀원의 피드백을 반영하여 애니메이션 속도와 조작 타이밍을 조정, 보다 직관적인 전투 흐름 설계 중요성을 고려하게 됨

→ QA가 필요한 이유와, 실질적인 테스팅 조정의 필요성 이해

05. StartRail

https://github.com/kiwibird0417/StartRail
https://cafe.naver.com/ictacademy/12959





진행 기간 2024.12.20~2025.01.22(5주)

소개 붕괴 스타레일 턴제 게임 모작

개발 규모 2인

담당 파트

기획 : 플레이어, 에너미 데이터 테이블 작성

개발 : 전체 게임 UI 연결 및 애니메이션 적용

구현 환경

엔진: 언리얼

관리: GITHUB, 구글 스프레드시트

개발환경: 블루프린트

05. StartRail - 기획 의도

https://github.com/kiwibird0417/StartRail
https://cafe.naver.com/ictacademy/12959

- 턴제 전투의 전략적 요소 강화

  - 턴 주도권 시스템 구현 → 적의 약점을 활용한 '약점 격파' 및 필살기 시스템을 통해 전투 흐름을 동적으로 변화

  - 턴 외 개입 시스템 반영 → 정적인 턴제가 아닌, 플레이어와 적이 서로의 턴에도 영향을 줄 수 있도록 설계

- 속성과 약점 기반 전투 시스템 구현

  - 속성 및 약점 데이터베이스 구축 → 아군과 적군이 공용으로 활용할 수 있는 데이터 시스템 설계

  - 전투 시스템 연동 → 데이터베이스 기반으로 적의 약점을 공격했을 때 턴을 빼앗는 메커니즘 구현

- 전투 흐름의 전략적 조절

  - 전투 흐름 조정 → 필살기 발동 시 턴을 활용한 전략적 변수 추가

  - 턴제 특유의 긴장감 유지 → 단순한 공격 교환이 아닌, 지속적인 개입과 변수 조작을 통한 플레이어의 선택 확장

05. StartRail - 데이터 테이블 및 UI

https://github.com/kiwibird0417/StartRail
https://cafe.naver.com/ictacademy/12959

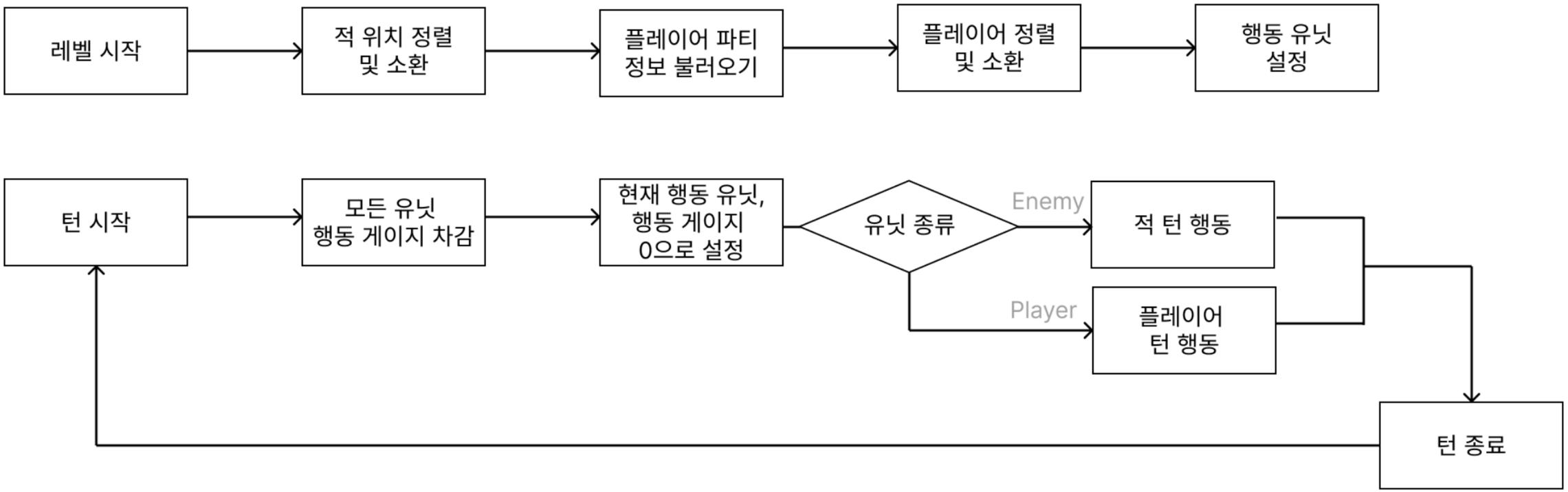
- 데이터 기반 전투 설계
  - 전투 데이터 테이블 구축 → 속성, 체력, 약점, 행동 패턴 등의 데이터를 정리하여 전투 시스템에 반영
  - 공식 자료 및 커뮤니티 조사 → 스타레일의 속성 시스템과 약점 격파의 영향을 분석하여 데이터 구조화
  - 속성 상성과 추가 효과 반영 → 빙결 시 행동 지연, 화염 시 지속 피해 등 속성 효과를 고려한 데이터 설계

	Row Name	Name	Path	Type	Weakness1	Weakness2	Weakness3	Level	HP	Attack	Defense	Speed	Taunt
1	Default	Default(Temp)	None	None	None	None	None	1	100	100	100	100	100
2	Trailblazer	Trailblazer	The Destruction	Physical	None	None	None	1	1203	620	460	100	125
3	Gepard	Gepard	The Preservation	Ice	None	None	None	1	1397	543	654	92	150
4	Bronya	Bronya	The Harmony	Wind	None	None	None	1	1241	582	533	99	100
5	BubbleHound	Bubble Hound	None	Physical	None	None	None	1	633	65	210	100	0
6	SweetGorilla	Sweet Gorilla	None	Ice	None	None	None	1	5064	65	210	143	0
7	DanHeng	DanHeng	The Hunt	Wind	None	None	None	1	882	546	396	110	75
8	March7th	March7th	The Preservation	Ice	None	None	None	1	1058	511	573	101	150
9	Natasha	Natasha	The Abundance	Physical	None	None	None	1	1164	476	507	98	108

- 직관적인 UI 설계
  - 속성 약점 표시 개선 → 적의 약점을 쉽게 인식할 수 있도록 아이콘과 색상 강조
  - 턴 순서 시각화 → 플레이어가 다음 행동을 예측할 수 있도록 턴 순서를 UI로 표시
  - 버튼 배치 최적화 → 자주 사용하는 공격 및 스킬 버튼을 직관적으로 배치하여 조작 편의성 강화

05. StartRail - 턴제 전투 플로우

https://github.com/kiwibird0417/StartRail
https://cafe.naver.com/ictacademy/12959



05. StartRail - 결과 및 회고

https://github.com/kiwibird0417/StartRail
https://cafe.naver.com/ictacademy/12959

1. 개발 과정에서의 어려움과 해결 방법

- 데이터 테이블을 활용한 전투 구조화:
  - 팀원이 설계한 전투 시스템을 분석하며 로직을 이해하는 데 집중했으나, 직접 구현하지 못한 점이 아쉬움으로 남음
  - 팀원이 구현한 전투 시스템에 맞도록 데이터 테이블을 구조화하여 전투 설계를 지원함

- 전투 확장성과 유지 보수성 부족:
  - 복잡한 전투 로직을 기획적으로 조율하는 과정에서 한계를 경험

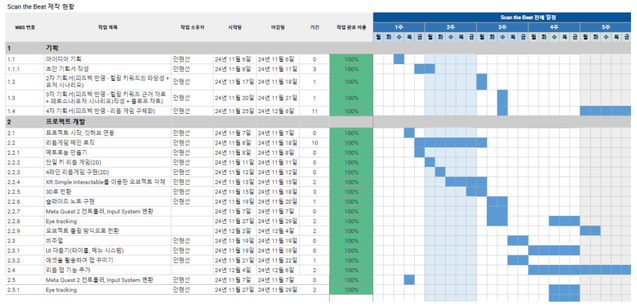
2. 피드백 및 개선 방향

- 전투 UI 개선 필요
  - 턴 순서 표시와 속성 강조를 통해 직관성을 높였으나, 추가 전투 요소 반영이 어려운 구조로 설계됨

- 시스템 설계의 중요성
  - 단순한 정보 전달이 아닌, 플레이어의 즉각적인 판단을 유도하는 시스템 설계가 전투에서 중요한 요소임을 깨달음

06. Scan the Beat

https://github.com/kiwibird0417/Scan-the-beat
https://drive.google.com/file/d/1pLjoiZiGqyqDUCAKVFhzo35I1FyLCO5Y/view?usp=sharing



Scan the Beat 제작 현황

WBS 번호	작업 제목	작업 소유자	시작일	마감일	기간	작업 완료 비율
1	기획					
1.1	아이디어 기획	안현선	24년 11월 6일	24년 11월 6일	0	100%
1.1.1	초안 기획서 작성	안현선	24년 11월 6일	24년 11월 11일	3	100%
1.2	2차 기획서(피드백 반영 - 힙합 키워드의 타당성 + 음악 시너지)	안현선	24년 11월 17일	24년 11월 18일	1	100%
1.3	3차 기획서(피드백 반영 - 힙합 키워드 근거 자료 + 레트로사운드 시너지요지성 + 플로우 자료)	안현선	24년 11월 20일	24년 11월 21일	1	100%
1.4	4차 기획서(피드백 반영 - 리듬 게임 구체화)	안현선	24년 11월 25일	24년 12월 6일	11	100%
2	프로젝트 개발					
2.1	프로젝트 시작, 깃허브 연동	안현선	24년 11월 7일	24년 11월 7일	0	100%
2.2	리듬게임 핵심 로직	안현선	24년 11월 8일	24년 11월 18일	10	100%
2.2.1	메트로놈 만들기	안현선	24년 11월 8일	24년 11월 8일	0	100%
2.2.2	만들 키 리듬 게임(2D)	안현선	24년 11월 11일	24년 11월 11일	0	100%
2.2.3	4차인 리듬게임 구현(2D)	안현선	24년 11월 12일	24년 11월 12일	0	100%
2.2.4	XR Simple Interactable를 이용한 오브젝트 삭제	안현선	24년 11월 13일	24년 11월 15일	2	100%
2.2.5	3D로 전환	안현선	24년 11월 15일	24년 11월 18일	3	100%
2.2.6	슬라이드 노트 구현	안현선	24년 11월 19일	24년 11월 20일	1	100%
2.2.7	Meta Quest 2 컨트롤러, Input System 변환		24년 11월 7일	24년 11월 7일	0	100%
2.2.8	Eye tracking		24년 11월 27일	24년 11월 29일	2	100%
2.2.9	오브젝트 풀링 방식으로 전환		24년 12월 2일	24년 12월 4일	2	100%
2.3	마무리	안현선	24년 11월 19일	24년 11월 19일	0	100%
2.3.1	UI 마들기(타이틀, 메뉴 시스템)	안현선	24년 11월 19일	24년 11월 19일	0	100%
2.3.2	애셋을 활용하여 맵 꾸미기	안현선	24년 11월 21일	24년 11월 22일	1	100%
2.4	리듬 만 기능 추가		24년 12월 4일	24년 12월 6일	2	100%
2.5	Meta Quest 2 컨트롤러, Input System 변환	안현선	24년 11월 7일	24년 11월 7일	0	100%
2.5.1	Eye tracking	안현선	24년 11월 27일	24년 11월 29일	2	100%

Scan the Beat 전체 일정

1주 2주 3주 4주 5주

월 화 수 목 금 월 화 수 목 금 월 화 수 목 금 월 화 수 목 금 월 화 수 목 금

진행 기간 2024.11.7~2024.11.30(4주)

소개 바코드 스캐너 VR 리듬 게임

개발 규모 2인

담당 파트

기획: 리듬 게임

개발: BPM에 맞춘 노트 생성, 컨트롤러를 활용한 노트 판정

구현 환경

엔진: 유니티

언어: C#

개발환경: VSCODE

ETC. GITHUB, VR

06. Scan the Beat - 기획 의도

https://github.com/kiwibird0417/Scan-the-beat
https://drive.google.com/file/d/1pLjoiZiGqyqDUCAKVFhzo35I1FyLCO5Y/view?usp=sharing

- VR의 몰입도 극대화

  - "현실적인 물체와 가상의 상호작용"을 핵심 개념으로 설정하여, 기존 VR 리듬 게임과 차별화

- 바코드 스캐너를 활용한 조작 방식

  - 단순한 버튼 입력이 아닌, 플레이어가 실제로 스캔하는 듯한 경험 제공

  - VR 컨트롤러의 특성 적극 활용

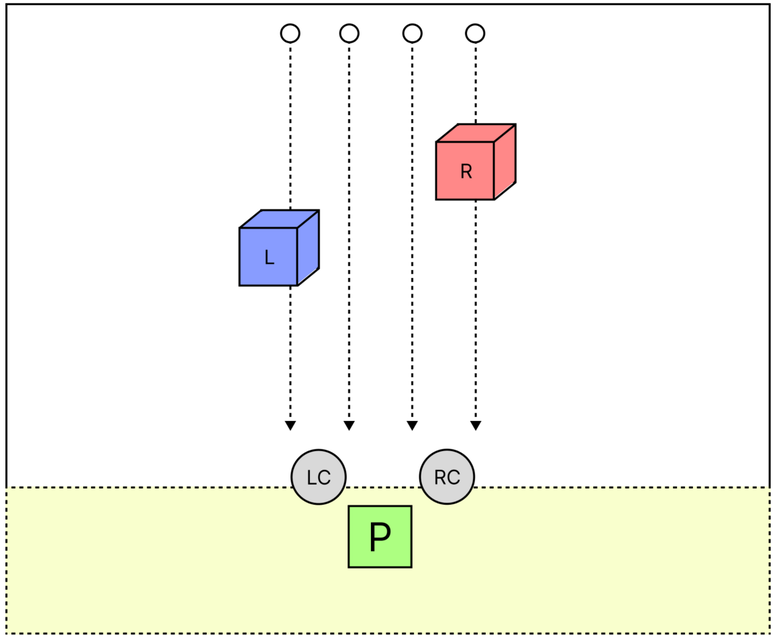
    → 비트 세이버의 손을 휘두르는 방식에서 벗어나, 보다 직관적이고 현실적인 리듬 액션 설계

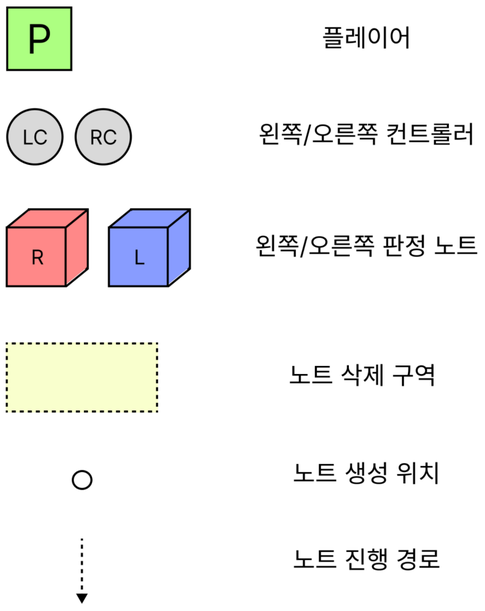
- 비현실적인 환경 속 현실적인 상호작용 구현

  - 실제 바코드 인식 기능은 없지만, 현실적인 조작감을 통해 게임 내 오브젝트와의 상호작용을 자연스럽게 유도

06. Scan the Beat - 게임 로직

https://github.com/kiwibird0417/Scan-the-beat
https://drive.google.com/file/d/1pLjoiZiGqyqDUCAKVFhzo35I1FyLCO5Y/view?usp=sharing



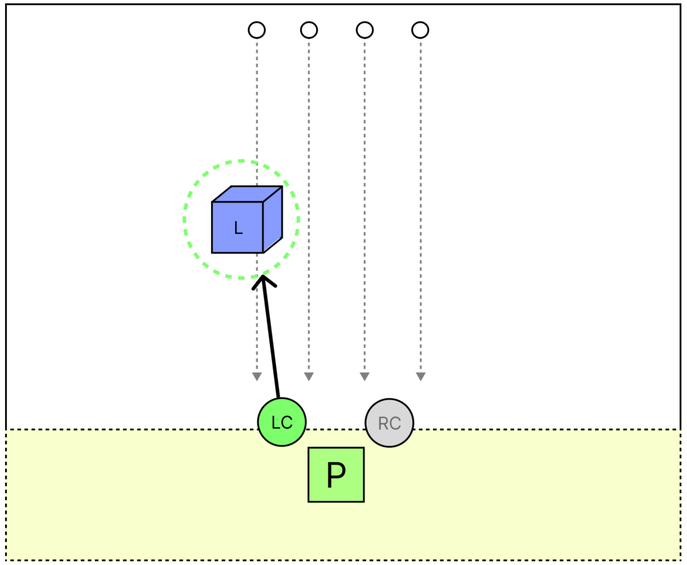


플레이어는 맵의 중심에 생성된다.

미리 생성된 리듬 패턴에 따라 노트 생성 위치에서 노트가 생성된다.

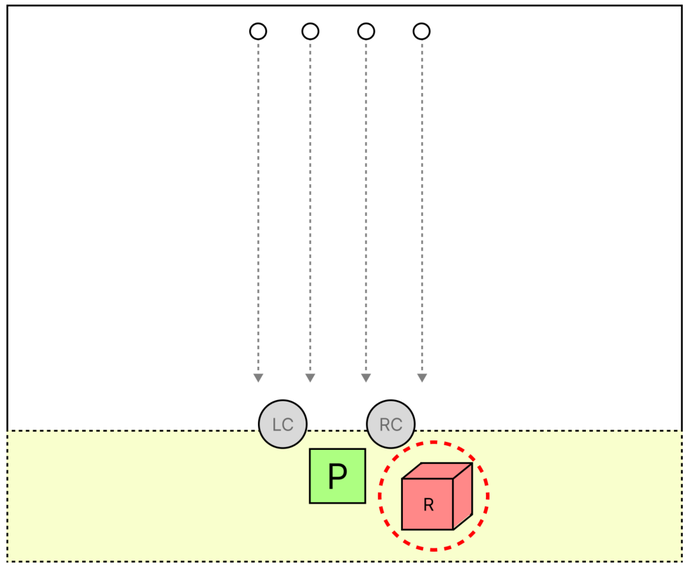
06. Scan the Beat - 게임 로직

https://github.com/kiwibird0417/Scan-the-beat
https://drive.google.com/file/d/1pLjoiZiGqyqDUCAKVFhzo35I1FyLCO5Y/view?usp=sharing



노트에 상응하는 컨트롤러 Trigger Action

실행 시 노트 성공 판정이 되어 점수가 올라간다.

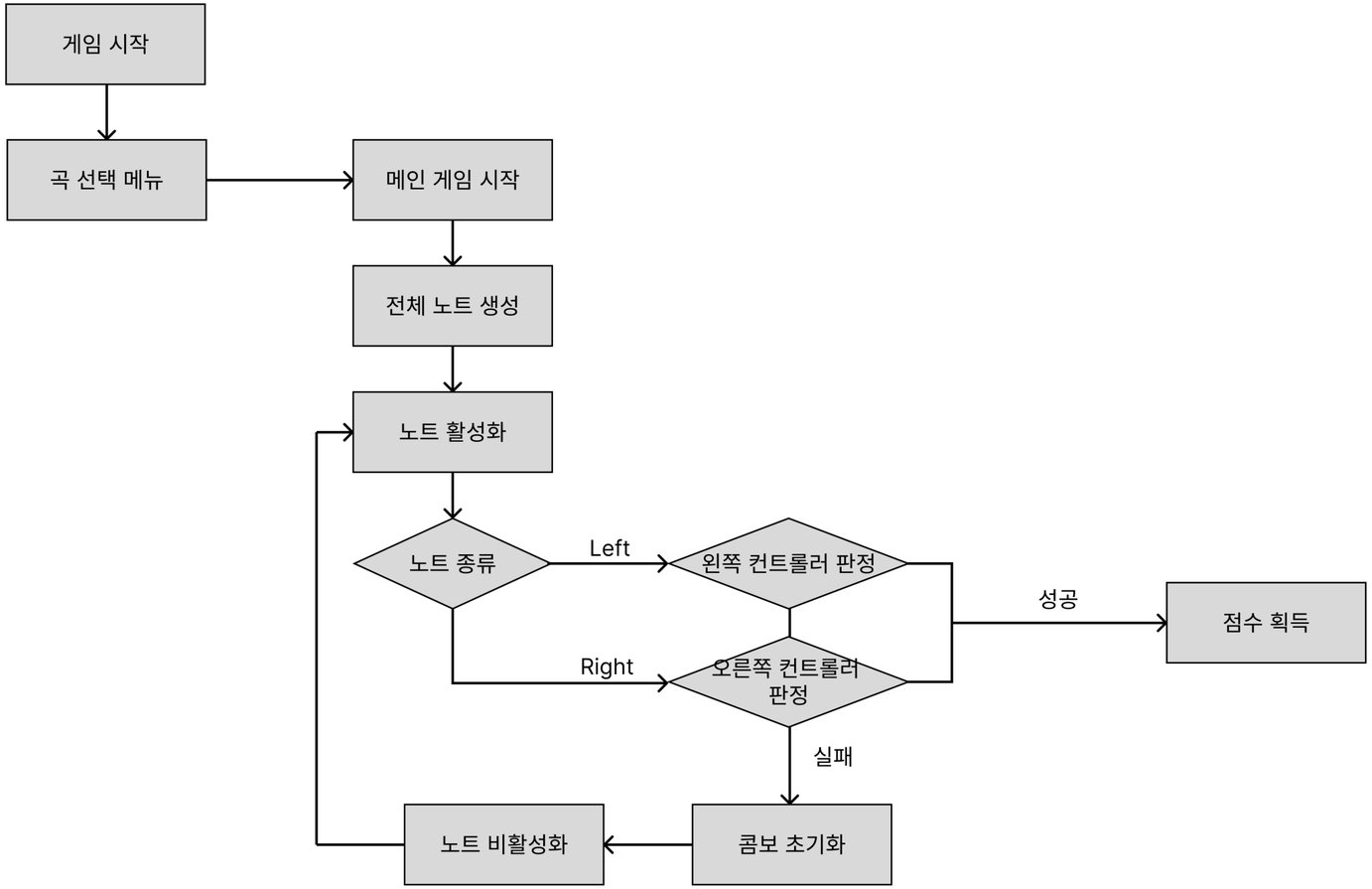


노트가 플레이어 중심을 지나, 삭제 구역에

들어가면 Object Pooling으로 비활성화된다.

06. Scan the Beat - 플로우 차트

https://github.com/kiwibird0417/Scan-the-beat
https://drive.google.com/file/d/1pLjoiZiGqyqDUCAKVFhzo35I1FyLCO5Y/view?usp=sharing



06. Scan the Beat - 결과 및 회고

https://github.com/kiwibird0417/Scan-the-beat
https://drive.google.com/file/d/1pLjoiZiGqyqDUCAKVFhzo35I1FyLCO5Y/view?usp=sharing

1. 개발 과정에서의 어려움과 해결 방법

- 타격감 및 피드백 부족 문제:
  - 리듬 게임에서 중요한 타격감과 즉각적인 반응 피드백이 부족하여, 시각적·청각적 피드백을 추가하려 했으나 개발 기간 내 충분한 테스트를 진행하지 못함

- VR 조작 방식의 몰입감 강화:
  - 현실적인 기기 활용이 신선한 경험을 제공했으며, VR 컨트롤러의 특성을 적극적으로 활용하여 조작감을 높임

2. 피드백 및 개선 방향

- 감각적 피드백 강화 필요:
  - VR에서는 단순한 인터랙션만이 아니라, 타격감·진동·시각적 효과 등의 피드백을 통해 몰입감을 극대화하는 것이 중요

- 반복적인 테스트와 조정의 중요성 인식:
  - 향후 프로젝트에서는 초기 기획 단계에서부터 다양한 피드백 시스템을 설계하고, 충분한 테스트를 통해 조작의 완성도를 높이는 방향으로 개선