

Portfolio



이세철

李世哲, Lee Sei Cheol

생년월일: 1999.12.30

번호: 010-4015-9749

E-mail: seichulee@naver.com

주소: 경기도 안양시

최종 학력: 2018.02 - 부흥고등학교 졸업

2024.02 - 한국공학대학교

(구 한국산업기술대학교) 게임공학과 졸업예정

Notion - [기획 포트폴리오 \(notion.site\)](https://notion.site)

Github - <https://github.com/kiption?tab=repositories>

목차

01 • 프로필

3p

- 수강 과목
- Skill
- Team Project

02 • 팀 프로젝트

5p

- Revenger – 소개
- Revenger – 기획
- Revenger – 개발(서버)
- Revenger – 개발(클라)

03 • 개인 프로젝트

17p

- Executioner_Record
- Token of Star

04 • 교내 활동

31p

- 공부 소모임
- 게임공학과 운영기획부

01

프로필

01. 프로필

주요 수강 과목

- 2021
 - 게임 기획 1
 - 컴퓨터 그래픽스 (OpenGL)
 - 자료구조 (C/C++)
- 2022
 - 게임 기획 2
 - 기획 포트폴리오
 - 스크립트 언어 (Python/OpenAPI)
 - 3D 게임 프로그래밍 1 (DirectX)
 - 네트워크 게임 프로그래밍 (TCP)
- 2023
 - 게임 엔진 1 (Unity)
 - 게임 서버 프로그래밍 (IOCP)

Skill

- 문서
PowerPoint / Excel / Word
- 개발
C++ / OpenGL / DirectX12 /
GitHub / Python / Unity /
C# / IOCP / AutoHotKey
- DB
MySQL / MSSQL

Team Project

- Revenger
IOCP/DirectX 12/C++
- Racer
TCP/DirectX 12/C++
- WhereFi
Python/Open API

02

팀 프로젝트

02. 팀 프로젝트

Revenger - 소개

REVENGER	
장르	TPS/FPS, 액션
플레이 인원	3명
플레이 타임	5~7분
게임 설명	다른 플레이어들과 함께 제한 시간 안에 적 NPC를 모두 처치하고 거점 지역을 점령하는 멀티 게임
개발 인원	3인 개발 (클라이언트 1, 서버 1, 기획 1)
개발 환경	DirectX 12, IOCP, C++
개발 기간	2022.11 ~ 2023.07
맡은 업무	맵 배치 / 레벨 디자인 / 적 NPC AI / UI / 물리 충돌 로직

02. 팀 프로젝트

Revenger - 기획



[그림 3] Revenger 표지

Revenger

- “배틀 필드4”와 유사한 장르의 게임
- 3명의 팀원으로 많은 적을 상대
- 공중전과 지상전을 체험할 수 있도록 설계
- 피격 시 부위 별 각기 다른 연출

노션 다운 링크: <https://www.notion.so/Revenger-a02f401c616d463ba874d0e3c54fa4a5?pvs=4>

02. 팀 프로젝트

Revenger - 소개



[그림 1] 작품에 대한 스테이지 1의 인-게임 화면

헬기 플레이어 1인, 군인 플레이어 2인으로 맵 안에 있는 모든 적 NPC를 처치하는 게임.

적 NPC로는 헬기, 군인 종류가 있으며 적 헬기는 아군 헬기에게 데미지를 더 받고, 아군 군인에게는 데미지를 덜 받게 된다.

정해진 적 NPC의 수를 제거하면 지정된 구역에서의 점령이 해금된다.

점령을 100%하게 되면 해당 게임은 Clear로 종료된다.

이때, 각 구역마다 일정 체력을 회복하는 힐-팩이 존재하며, 자신의 체력이 0이 되는 순간 사망 판정으로 Game Over로 종료된다.

02. 팀 프로젝트

Revenger - 기획

목차

1. 게임 개요	2. 헬기 플레이어 설정	3. 군인 플레이어 설정	4. Stage 1 적 설정
1) 게임 소개	1) 손상도 시스템 정의	1) 군인 플레이어 상태개요	1) Stage 1 적 상태개요
2) 게임 목표	2) 헬기 플레이어 상태개요	2) 군인 플레이어 피격 판정영역	2) Stage 1 적 피격 판정영역
3) 전체 플로우차트	3) 헬기 플레이어 피격 판정영역	3) 군인 플레이어 HP 설정	3) Stage 1 적 내구도 설정
	4) 헬기 플레이어 내구도 설정	4) 군인 플레이어 조작 방법	
	5) 헬기 플레이어 조작 방법		
5. 맵 설정	6. 화면 설정	7. 연구과제 시나리오	8. 출처
1) Stage 1 지형 컨셉	1) 화면 구성		
2) Stage 1 구성	2) 로그인 화면		
3) Stage 1 배치	3) 로비 화면		
	4) 1스테이지 게임 UI 구성		

2

[그림 4] 작품의 세부 기획서 중 목차

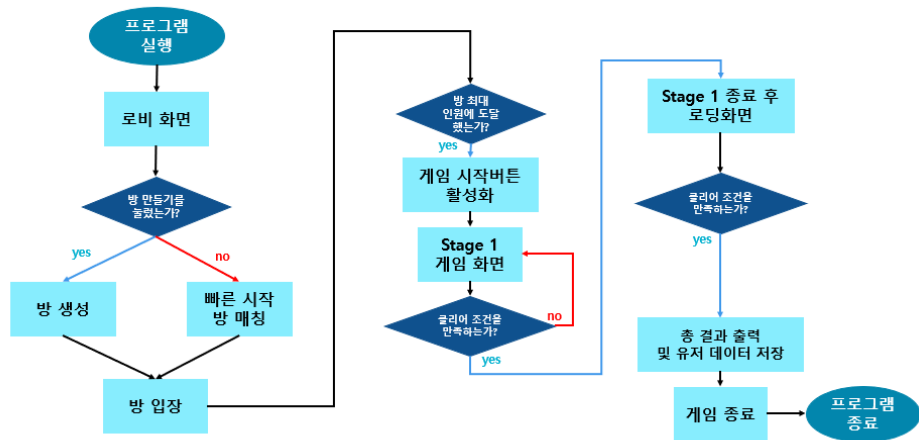
- 왼쪽 사진은 기획서 내용 중 목차이며, 후술할 내용은 해당 기획서를 **작성한 방법**입니다.
- 작품 개발에 필요한 게임 컨셉과 목표를 먼저 세웠습니다.
- 게임 내에 들어갈 시스템과 맵을 구상하여 작성하였습니다.
- 게임 내 시스템과 맞춰 플레이어, 적과 관련한 밸런스 및 조작을 작성하였습니다.
- 실제 화면에서 보일 UI/UX를 구상하여 작성하였습니다.

02. 팀 프로젝트

Revenger - 기획

1. 게임 개요

1.3. 전체 플로우차트



[그림 5] 전체적인 게임 도식화

- 왼쪽 사진은 전체 플로우 차트입니다.
- 프로그램 실행 시 로비로 진입하며 해당 화면과 관련해서 UI/UX를 제작하였습니다.
- 서버와 같이 작업하여 방 매칭 시스템을 구현하였습니다.
- 게임 시작 버튼 활성화가 되기 위해서는 방 인원이 최대이면서 모두가 역할을 선택 후 준비 상태가 되어야 합니다.
- 게임 종료 시 프로그램이 자동으로 종료됩니다.

02. 팀 프로젝트

Revenger - 기획

2. 헬기 플레이어 설정

2.3. 헬기 플레이어 피격 판정영역

- 각 부위 별 효과 연출은 다음과 같다.

발상 구역	프로펠러	부위 내구도 100	부위 내구도 40: 과열로 인한 이동속도 저하, 0: 프로펠러 파괴로 추락
	효과 연출	내구도 40일 때, 화면 테두리로 빨간 이펙트(사인선 같은) 출력과 프로펠러에서 이동경로를 그리면서 연기 출력 내구도 0일 때, 프로펠러 터지는 연출과 함께 푸쉬인 소리 출력, 조종하던 헬기는 조종 불가로 변형, 카메라는 공중에서 정지 카메라가 공중에서 멈춰 있을 때 헬기는 원래 이동방향대로 추락하며 땅에 닿을 시 폭발 이후, 회색화면과 함께 사망 다이얼 출력	
노량 구역	머리	부위 내구도 100	부위 내구도 40: 유리창 균열 효과, 0: 유리창 깨짐 효과 및 조준점 제거
	효과 연출	내구도 40일 때, 화면 테두리로 균열 효과 출력 내구도 0일 때, 화면 테두리로 깨진 유리 효과와 퍼진 균열 효과 출력 및 조준점 제거	
보라 구역	꼬리	부위 내구도 100	부위 내구도 40: 기체 회전 속도 증가, 0: 기체 정지 불가 및 지속적인 회전
	효과 연출	내구도 40일 때, 헬기의 이동경로를 그리는 연기 꼬리에서 출력, 회전 민감도 증가 내구도 0일 때, 꼬리부분 파괴(꼬리 부위 두 동강 및 폭발)과 카메라 공중에서 정지, 또한 카메라가 공중에서 멈춰 있을 때 헬기는 계속 y축을 기준으로 회전하며 추락 땅에 닿을 시 폭발	
연두 구역	몸통	부위 내구도 100	부위 내구도 40: 기체 받는 데미지 증가, 0: 기체 파괴로 인한 사망
	효과 연출	내구도 40일 때, 헬기 몸체 곳곳(몸체 내에서 랜덤 위치)에서 연기 발생(위로), 데미지 계산식 중 모든 부위가 받는 20% 데미지 추가 내구도 0일 때, 몸체가 두 동강나며 폭발	

[그림 6] Stage 1 플레이어 피격 시 부위 별 연출 상황

2. 헬기 플레이어 설정

2.4. 헬기 플레이어 내구도 설정

- 기체는 부위 별 내구도가 존재한다.

- 각 계산에 의한 데미지는 부위 별로 계산 됨
Ex) 대상과의 거리가 2Km이고, 피격 대상의 프로펠러의 내구도가 50인 상태에서 맞은 경우
→ 데미지 = $(20 * (20 * 5)) / 100 = 20$,
공격당한 대상의 프로펠러의 내구도는 $50 - 20 = 30$, 따라서 나머지 내구도는 50이지만, 프로펠러 내구도만 30이 된다.

플레이어 헬기	Apache
부위 별 내구도	각 100
헬기 방어력	방어력 50
기본 공격력	공격력 20, 1초에 두 발, 최대 탄창 100발
특수 능력	클타임 - 1분 30초 유도 미사일 5개가 줄지어서 사출 (발 당 고정 데미지: 10)
최대 이동 속도	1초 당 40M 전진
데미지 계산 식	데미지 = $(\text{공격력} * \text{대상과의 거리}(100\text{M} \text{ 당 } 5, \text{ 최대 거리는 } 2\text{km})) / \text{피격 대상의 방어력}$

- 미사일은 미사일 키 누르고 마우스 클릭 시 발생되며, 사출될 때 미사일의 꼬리 부분에서 연기가 출력되며 전진한다.
- 미사일은 지정된 방향으로 5km 날아가고 부딪힌 대상이 없으면 공중에서 폭발한다.
- 폭발 시 폭발 범위($1\text{km}^2 * 3 \text{ pi}$)

12

[그림 7] Stage 1 플레이어 스테이터스

위 기획은 Stage 1에 해당하며, 플레이어는 피격 부위가 총 4개가 존재합니다.

따라서 부위의 내구도 수치 별로 연출 상황이 다르게 설정하였습니다.

Stage 1의 플레이어는 헬기를 사용하기 때문에 사용할 기체 중 하나에 대해서만 능력치를 작성하고 플레이어가 피격 당했을 때의 데미지 계산 식을 작성하였습니다.

02. 팀 프로젝트

Revenger - 기획

5. 맵 설정

5.3. Stage 1 배치



맵

- Stage 1 : 도심지
- 약 $3km^2$ 의 크기
- 높은 건물들이 다수 배치
- 나무와 같은 환경 오브젝트들이 다수 존재.

오브젝트

- 여러 개의 오브젝트들
 - 플레이어(헬기 기체)
 - 장애물(나무, 고층 건물 등)
 - 적 종류 중 2개
 - 고층 건물 존재 (etc: 마천루)

30

[그림 8] Stage 1 맵 설계

- 해당 맵은 Stage 1로 도시에서 펼쳐는 전투라는 분위기를 내고자 하였습니다.
- 맵 곳곳에는 높은 건물들이 존재합니다.
- 나무, 차와 같은 작은 오브젝트들도 존재합니다.
- 게임 내 오브젝트들은 총알이 관통하지 못합니다.

02. 팀 프로젝트

Revenger - 기획

항목	1월	2월	3월	4월	5월	6월	7월	8월	9월
리소스 수집									
서버-클라이언트 프레임워크									
서버-클라이언트 통신 및 동기화									
애니메이션, 그림자, 파티클									
서버 이중화									
NPC 인공지능									
특수 능력, 손상 및 파괴 로직									
UI, 충돌처리, 모델 링크, 사운드									
오프닝									
릴리즈 준비 및 발표									

초기 계획

완료

NPC 충돌처리가 제대로 되지 않는
현상으로 인한 작업이 지연되었습니다.

이는 장애물 충돌 정보와
A* 알고리즘의 그래프 초기 설정이
서로 매칭되지 않았습니다.

따라서, 충돌 정보들에 대한 자원은
공유하고, A* 알고리즘 그래프 초기
설정

해당 프로젝트에 대해 해야 하는 작업과 달성율을 체크를 담당하였습니다.
팀 전체적으로 작업을 하기 때문에 초기 계획에서 정한 기간보다 더 늦게 끝나는 이슈들이 있었습니다.
말았던 업무 중 UI, 충돌 처리가 플레이어는 되지만 적 NPC에 적용되지 않는 이슈 같은 것들이 존재하였습니다.
제가 한 분야는 빨간색 경계 구간입니다.

02. 팀 프로젝트

Revenger - 개발(서버_프로그래밍한 내역)

```
class OBJECT {
public:
    mutex obj_lock;

    int id;
    char name[NAME_SIZE];
    short type;

    short state; // PLAYER_STATE
    int hp;
    int remain_bullet;

    XMFLOAT3 pos; // Position (x, y, z)
    float pitch, yaw, roll; // Rotated Degree
    XMFLOAT3 m_rightvec, m_upvec, m_lookvec; // 현재 Look, Right, Up Vectors

    BoundingBox m_xobb; // Bounding Box
```

[그림 10] NPC Class Code 中 일부

```
void NPC::NPC_State_Management(int state)
{
    switch (type)
    {
        case NPC_HELICOPTER:
        {
            switch (m_state)
            {
                case NPC_IDLE: {
                    // ....
                }
                case NPC_ATTACK: {
                    // ....
                }
                case NPC_DEATH: {
                    // ....
                }
                default:
                    break;
            }
            break;
        }
    }
}
```

[그림 11] 유한 상태 기계 함수: ST1_State_Management의 내용

서버에서 NPC가 상태에 따라 행동하도록 유한 상태 기계의 형태로 구현하였으며,
Lua 스크립트가 아닌 C++의 형태로 프로그래밍 되어있습니다.
현재 Class 형태로 NPC가 할 수 있는 모든 행동을 관리하고 있습니다.

02. 팀 프로젝트

Revenger - 개발(서버_프로그래밍한 내역)

```
struct RaycastResult {  
    bool hit; // 충돌 여부  
    float distance; // 레이의 시작점에서 충돌 지점까지의 거리  
    XMFLOAT3 hitPoint; // 충돌 지점  
};
```

```
RaycastResult Raycast(const XMFLOAT3& rayOrigin, const XMFLOAT3& rayDirection,  
    const BoundingBox& box) {  
  
    // 로컬 공간에서 월드 공간으로 충돌 지점을 변환  
    XMStoreFloat3(&result.hitPoint, XMLoadFloat3(&result.hitPoint) +  
        XMLoadFloat3(&box.Center));  
  
    return result;  
}
```

[그림 12] RayCast.h 中 충돌 계산의 일부

NPC가 추적하는 대상인 플레이어에게 공격을 할 경우, 공격할 때 총알의 좌표를 업데이트를 해야합니다. 서버의 부하를 줄이기 위해 Raycast를 적용하면 총알의 좌표를 업데이트 하지 않고도 충돌 판정을 내릴 수 있게됩니다. 이는 DirectX 12의 내장 함수가 아닌 수학적 지식을 이용하여 공식을 토대로 직접 만들어 사용하였습니다.

02. 팀 프로젝트

Revenger - 개발(클라_프로그래밍한 내역)

```
// Chat UI
if (UI_Switch) {
    D2D_POINT_2F D2_ChatUI = { FRAME_BUFFER_WIDTH - 782.0f, (FRAME_BUFFER_HEIGHT / 2) - 145.5f };
    D2D_RECT_F D2_ChatUIRect = { 0.0f, 0.0f, 782.0f, 291.0f };
    m_pd2dDeviceContext->DrawImage(m_pd2dfxGaussianBlur[47], &D2_ChatUI, &D2_ChatUIRect);

    D2D_POINT_2F D2_InsertChatUI = { FRAME_BUFFER_WIDTH - 782.0f, D2_ChatUI.y + D2_ChatUIRect.bottom + 5.0f };
    D2D_RECT_F D2_InsertChatUIRect = { 0.0f, 0.0f, 782.0f, 69.0f };
    m_pd2dDeviceContext->DrawImage(m_pd2dfxGaussianBlur[48], &D2_InsertChatUI, &D2_InsertChatUIRect);
}
```

[그림 13] GameFramework.cpp 함수 중 일부

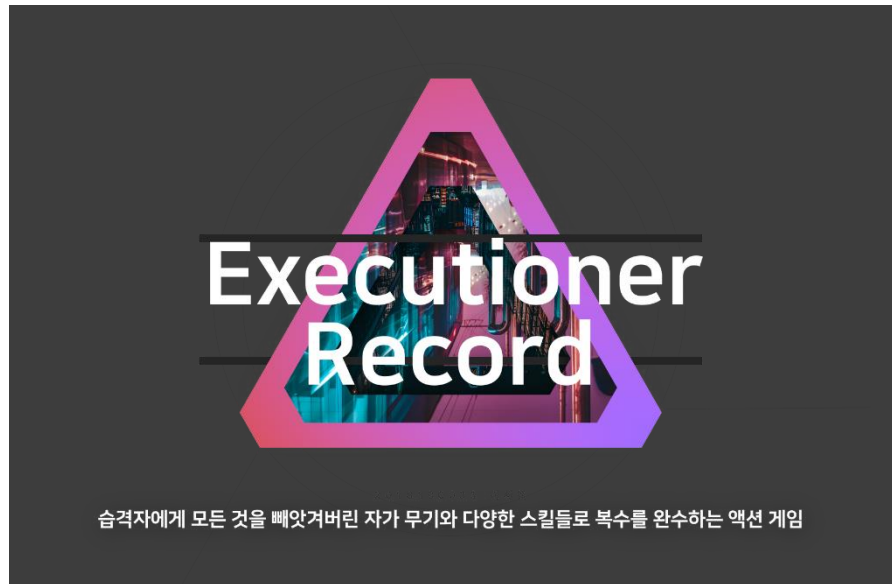
```
if (chat_comeTome) { // .... 자세한 내용은 코드 링크에 있습니다.
    if (nameLength < 80 - 2) { // tempText의 크기에서 ":"를 뺀 값보다 작은지 확인
        strcpy_s(tempText, sizeof(tempText) / sizeof(tempText[0]), name);
        strcat_s(tempText, sizeof(tempText) / sizeof(tempText[0]), ":" ); // ":" 추가
        strncpy_s(tempText, sizeof(tempText) / sizeof(tempText[0]), msg, _TRUNCATE);
        converttext = ConvertToWideChar(tempText);
        wcsncpy_s(textinfo.chatData, converttext);
        delete[] converttext;          gGameFramework.m_chat_info.push(textinfo);
    }
} // 7. 채팅 동기화
```

[그림 14] Network.h 함수 중 일부

DirectX에서 2D Device를 이용하여 여러 UI를 클라이언트의 Frame Work에서 띄우고 있습니다.
추가적으로 UI는 서버에서 연동하기 때문에, 서버에서 받아온 값을 이용하여 UI 출력 범위를 조정합니다.

03

개인 프로젝트



[그림 18] Executioner_Record 표지

Executioner_Record

- “고스트 러너” 게임과 유사한 장르의 게임
- 액션-파쿠르 장르에 RPG적 요소를 추가
- 무기에 따른 **다양한** 스킬
- 진행할수록 어려워지는 **맵 구조**
- 주변 아이템 및 장애물, 구조물과 다양한 **상호작용**을 할 수 있음

노션 다운 링크: [Executioner_Record \(notion.site\)](https://www.notion.site/Executioner-Record)

03. 개인 프로젝트

플레이
EXECUTIONER RECORD

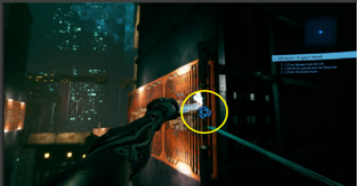
3.3 구조물 상호작용

플레이어는 맵 곳곳에 설치된 구조물들과 상호작용을 하여 이동을 할 수 있게 된다.

와이어 이동

벽에 있는 고리 부분에 플레이어는
와이어를 걸어 빠른 이동이 가능하다.

키보드 Q키 누른 상태에서 상호작용 포인트를
바라본 상태로 좌 클릭하게 되면
와이어가 나기며 이동이 가능해진다.



<그림 19> 와이어 이동 예시

20/65

[그림 19, 20] 구조물 상호작용에 대한 자세한 내용 및 상호 작용 방법

플레이
EXECUTIONER RECORD


3.3 구조물 상호작용

플레이어는 맵 곳곳에 설치된 나무상자를 열어 사용 아이템을 얻을 수 있다.

나무상자

스테이지 곳곳에 나무상자가 존재하며,
스테이지 별 나무상자 개수가 다르다.
상자 앞 위치로 가서 마우스 좌 클릭 할 때
상자가 열리게 된다.

연 상자의 아이템은 **자동적으로**
사용 아이템(4번) 창으로 들어가게 된다.

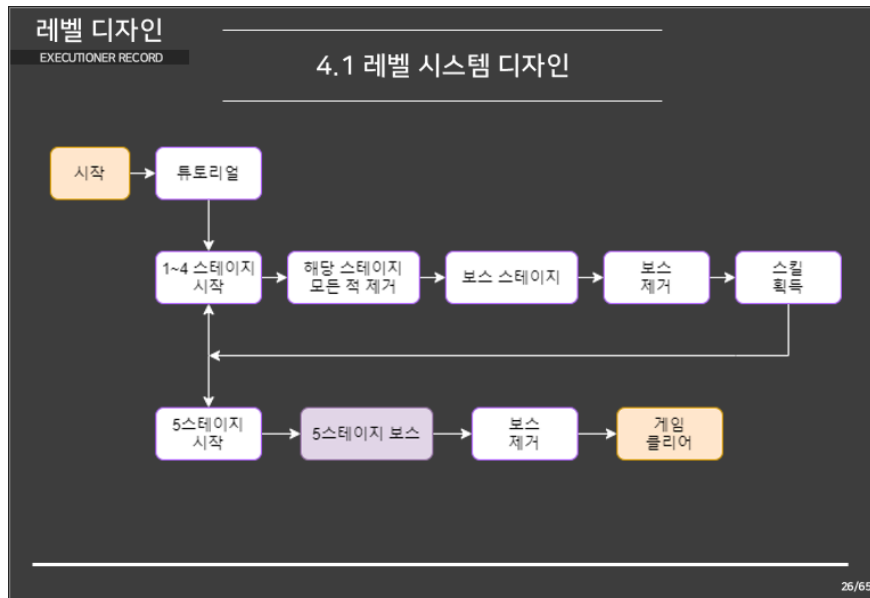


<그림 13> 스테이지 안-개입 내 나무상자 이미지 예시

24/65

구조물들과 상호작용을 하기 위해 특정 키를 설정하여 누르면 상호작용이 되도록 설정하였고,
각 구조물들과 상호작용 시 이동하거나 업뎀을 하는 등의 추가 행동을 하도록 설계하였습니다.

03. 개인 프로젝트



[그림 21] Executioner_Record에 대한 플로우 차트

레벨 디자인
EXECUTIONER RECORD

4.2 무기 시스템

<그림 17> 단검으로 공격하는 모습 예시

<그림 16> 장검으로 공격하는 모습 예시

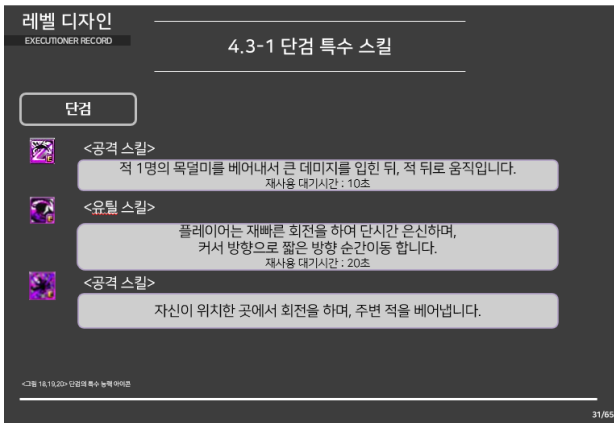
	단검	장검
공격속도	1초에 2번 휘두름.	1초에 1번 휘두름.
체력	기본 체력 200	기본 체력 230
이동속도	초당 1.5M(걸기)/ 초당 4M(뛰기)	초당 1M(걸기)/초당 3M(뛰기)
기본 공격 사거리	캐릭터로부터 1.5M	캐릭터로부터 2M

29/65

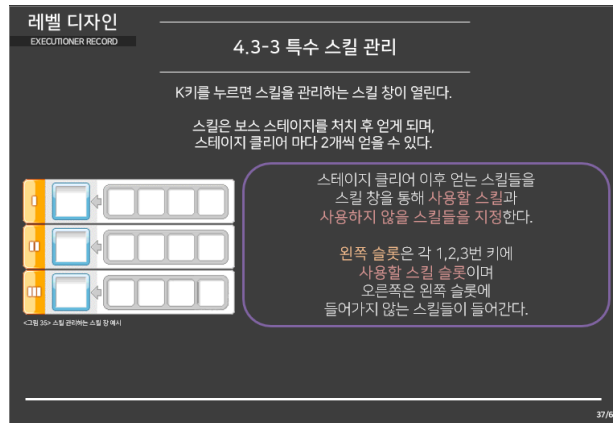
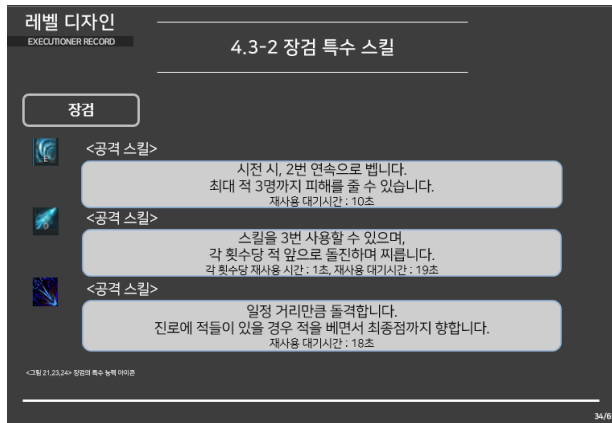
[그림 22] 무기 시스템에 대한 기본 설계

게임에 대한 전반적인 플로우 차트와 무기에 따른 다양한 스킬을 제공하기 위해 기본적인 무기들의 성능 표 및 시스템을 설계하였습니다.

03. 개인 프로젝트



[그림 23, 24] 단검과 장검의 스킬 리스트 中 일부



[그림 25] 스킬 선택 프리셋


튜토리얼 이후 선택하는 무기에 따라 다양한 스킬을 스테이지 통과할 때마다 얻을 수 있고, 각 무기 별 스킬 리스트를 작성하였습니다. 이때, 범위와 타격 횟수, 스킬 타입, 쿨타임 등을 설정했습니다. 다양한 스킬을 얻었을 경우 특정 스킬만 골라 사용하게끔 하는 UI를 설계했습니다.

03. 개인 프로젝트

레벨 디자인
EXECUTIONER RECORD

4.4 맵

1 스테이지



<그림 36> 1.5 스테이지 예시 화면

- 1 스테이지는 자신을 숨겨한 무리의 사원이 무대이다.
- 사원의 벽이나 지붕을 넘나 들어 적을 처치한다.
- 맵 크기는 약 1km * 1km 정도의 면적을 갖는다.
- 1 스테이지 맵 안 구급상자는 7개 존재한다. 회복되는 체력은 전체 체력의 40% 이다.
- 나무상자는 3개 존재한다. 습득한 아이템은 바로 사용아이템슬롯으로 옮겨지며, 전에 사용하던 아이템이 존재할 경우 새 아이템으로 갱신된다.

38/65

[그림 26, 27] 각 스테이지 간의 공통점 및 차별화되는 요소

레벨 디자인
EXECUTIONER RECORD

4.4 맵

1 스테이지

등급	패턴 종류	설명	데미지
일반	X	X	X
간부	기습	플레이어 방향으로 순간적인 <u>대쉬</u> 와 함께 검을 휘두른다. (1분 당 1회)	45
	암기	플레이어와 일정 거리(20m) 이상 떨어진 경우 암기를 던진다. (연속 3회, 최초 시전으로부터 <u>쿨타임</u> 1분)	1회당 5
보스	기습	전투 첫 진입 시 교주가 그림자에 숨어 3초 뒤 플레이어를 급습한다. (최초 1회만 시전)	즉사
	방어	절대 방공진을 시전하여 10초 간 무적이며, 해당 시간에 공격해서 해제하지 못하면 10초 뒤 즉사 데미지를 받는다. (플레이어의 타격수 30회 이상 시 방어진 해제, 전투 3번 사용, <u>쿨타임</u> 1분 30초)	즉사
	타격	땅에 검기를 날려 2초 뒤 5회 동안 땅 속에서 플레이어의 위치로 검기가 발사된다. (최초 시전으로부터 <u>쿨타임</u> 1분)	1회당 15

41/65

각 스테이지 맵의 배경과 스토리 라인을 따라 스테이지 별 미션 및 자연환경을 설정하고 장애물 배치, 특징 등을 표로 먼저 작성하였습니다.

03. 개인 프로젝트

추후 계획

EXECUTIONER RECORD

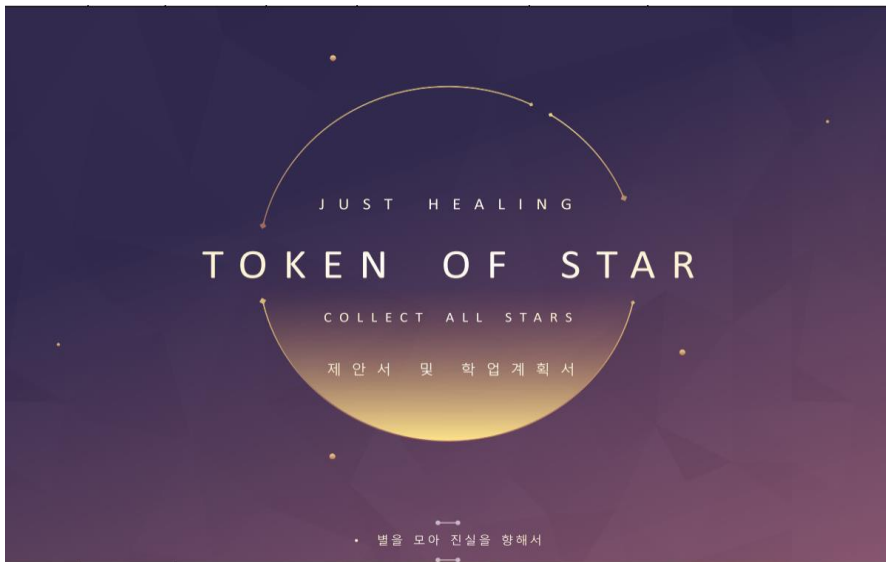
요소	내용
컨트롤	존재하는 더블 <u>대쉬</u> 나 더블 점프 등 다른 커맨드의 구체화
비고	적의 데미지, 공격속도, 사거리 등의 세부 사항들 설정
	다양한 유형 (예- 실드를 갖고 있는 적 등)의 특별한 능력을 지닌 적 설정.
	난이도 별 모드 설정 1회 차 플레이를 클리어 할 때, 여러 난이도의 모드를 설정 기록경쟁 할 수 있는 모드도 설정

63/65

[그림 28] 기획서에 대한 추가할 내용

- 전반적인 게임 시스템과 스토리는 완성되었다고 생각이 되어 해당 부분에 대한 추가적인 요소는 없습니다.
- 컨트롤이나 다양한 유형의 적과 난이도 설정이 필요하기에 추후 추가할 예정입니다.

03. 개인 프로젝트



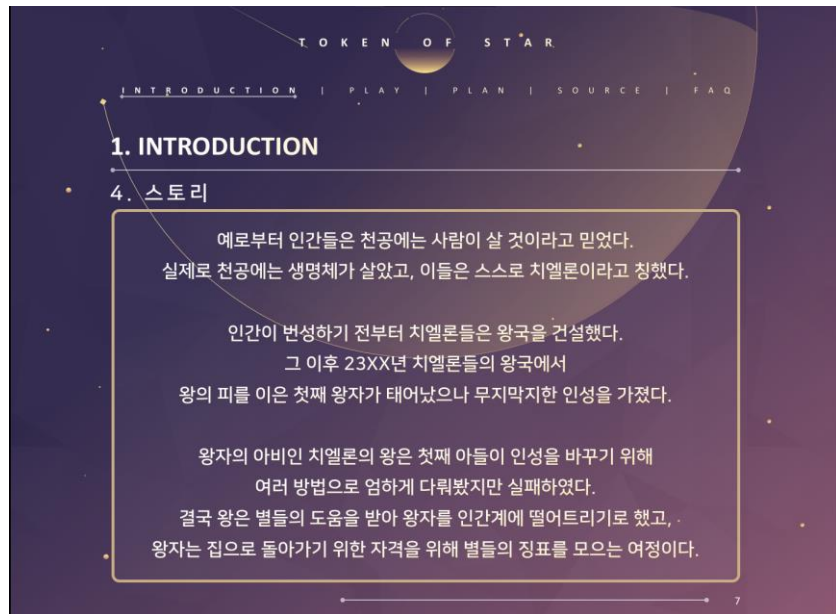
[그림 37] Token of Star 표지

Token of Star

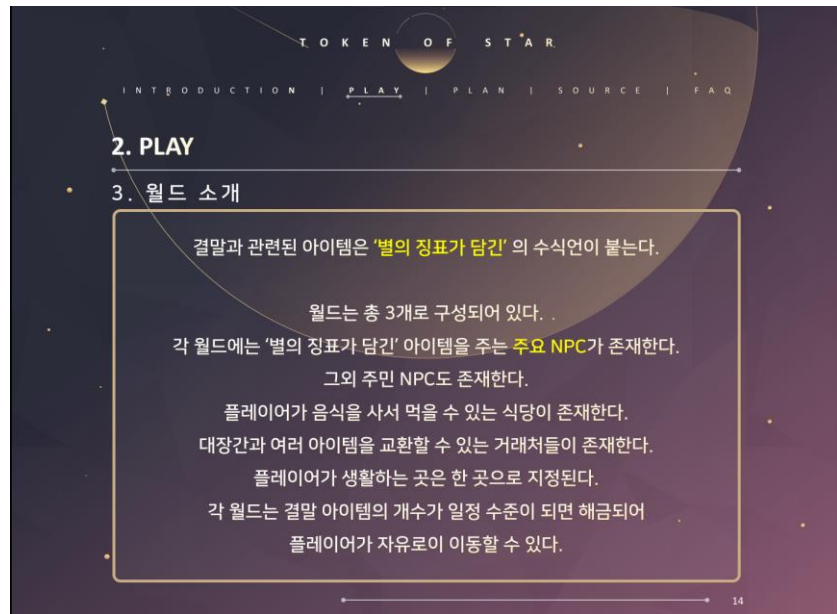
- "스타듀밸리"와 유사한 게임
- 파밍뿐만 아니라 아이템에 따라 **게임 스토리가 달라지는** 게임
- 전투, 생활 등의 시스템이 존재
- 퀘스트 및 호감도를 통해 다른 월드 개방
- **스토리 아이템**과 **명성**에 따라 변하는 엔딩

노션 다운 링크: [Token of Star \(notion.site\)](https://www.notion.site/Token-of-Star)

03. 개인 프로젝트



[그림 38] Token of Star 전반적인 스토리 내용



[그림 39] Token of Star 게임 내 월드와 구성요소 설명

스토리과 스토리 아이템을 연관지어 유저가 스토리에 몰입하고 게임을 즐길 수 있도록 설계하였습니다.
단순히 아이템을 얻고 끝나는 것이 아닌, 아이템을 얻기 위한 생존과 호감도 등을 완수해야 합니다.

03. 개인 프로젝트

기본 룰
※ 현실세계의 1초가 게임 세계의 30초. 즉, 1분은 30분이 흘러간 것이다.
※ 기상: 오전 8시, 취침: 최대 오전 12시.
※ 기상 이후 취침은 아무때나 가능하다.
※ 기상 후 오른쪽의 활동을 선택한다.
※ 플레이어 생성 시 기본 스탯으로 생성

기상 후 활동 선택 내역
1. 농활 활동 2. 자유 활동
▶ 농사 ▶ 수련
▶ 낚시 ▶ 호감도
▶ 퀘스트

플레이어 스탯		
	기본 스탯	최대 스탯
공격력	3	50
이동속도	50	150
활동력	70	200
HP	20	120

게임 플레이 화면 예시

1. 플레이어의 HP를 표시한다. (왼쪽이 떨어진 HP, 오른쪽이 전혀 떨어지지 않은 HP를 표시한다.)
2. 플레이어의 활동력을 표시한다. (왼쪽이 전혀 떨어지지 않은 활동력, 오른쪽이 떨어진 활동력을 표시한다.)
3. 현재 플레이어의 위치를 표시한다.
4. 농활활동의 경우 아래 사용 도구가 표시된다. (평상 시에는 없다.)
5. 현재 플레이어의 명성을 표시한다.
6. 게임 내 시간을 표시한다.

플레이어 정보 화면 예시

먼저 P키를 통하여 플레이어의 정보를 확인한다.

1. 플레이어의 이름을 표시한다.
2. 플레이어 활동에 의한 총 합 레벨을 표시한다.
3. 소유하거나 장착 중인 특수장비를 표시한다.
4. 커스터 마이징된 플레이어를 표시한다.

위 사진에서 여백 공간에 제력이나 공격력을 표시할 예정이다.
다른 공간에 활동에 의한 레벨이나 수련 레벨, NPC간의 호감도를 표시할 예정이다.

[그림 40] Token of Star 디폴트 설정 내역

생활은 3D “스타듀밸리”의 형태로 설정하였으며, 기본 스테이터스 및 플레이어 화면, UI 등을 플레이했던 게임에 덧붙여 작성하였습니다.

03. 개인 프로젝트

<퀘스트 주요 목차>

- ◆ NPC들의 심부름
- ◆ 스토리를 위한 주요 NPC의 연계 퀘스트
- ◆ 자유 활동에서 얻은 특정 퀘스트 아이템으로 부터 생성된 퀘스트 혹은 스토리와 관련이 없는 퀘스트

- ※ 모든 퀘스트는 제한시간이 정해져있다. 정해진 시간을 넘길 경우 퀘스트 실패
- ※ 주요 퀘스트를 제외한 서브 퀘스트는 수행 자유

<주요 퀘스트 목록>

- ◇ (수렵 퀘스트) 제한된 구역 내에서 농사활동
 - 특정 아이템 수렵하여 해당 NPC에게 전달하기
- ◇ (낚시 퀘스트) 제한된 구역 내에서 낚시활동
 - 특정 물고기 낚시하여 해당 NPC에게 전달하기
- ◇ (호감도 퀘스트) 퀘스트 부여된 월드 내에서 NPC들에게 말걸기
- ◇ (단서 퀘스트) 퀘스트 부여된 월드 내에서 특정 단서들을 시간 내에 연계하여 찾기
- ◇ (처치 퀘스트) 특정 구역으로 워프되어 짐승 혹은 몬스터 처치

[그림 41, 42, 43] Token of Star 퀘스트, 호감도, 명성

자유 활동

- ▶ 자유 활동인 상태에서 가능하다.
- ▶ 일상 인벤토리를 사용한다.

활동 내역

- ▷ NPC들과 대화를 한다.
- ▷ 원하는 음식이나 물건을 선물한다.
- ▷ NPC가 특정 호감도 상태에 도달하면 해당 NPC의 호감도가 MAX가 된다.
- ▷ NPC와 하루에 3번 일반 대화할 수 있다.
- ▷ 일반 대화는 대화 한 번당 활동력 10을 소모한다.
- ▷ NPC와 일주일에 한 번 특수 대화를 할 수 있다.
- ▷ 특수 대화는 한 번당 활동력 30을 소모한다.
- ▷ 일반 대화는 호감도 20을 올려준다.
- ▷ 특수 대화는 호감도 100을 올려준다.
- ▷ 특정 아이템을 선물할 경우 선물의 선호도에 따라 호감도가 50/30/10/-5를 얻는다.

명성

- ▷ 초기 상태인 0에서 시작한다.
- ▷ 퀘스트의 보상으로 명성을 선택하면 퀘스트마다 5~15의 명성을 얻을 수 있다.
- ▷ 호감도가 MAX인 NPC의 보상으로 받을 수 있다.
- ▷ 명성 수치가 50이 되면 '별의 징표가 담긴' 아이템을 얻는다.
- ▷ 명성 수치가 50이 되면 위의 아이템을 얻고 수치가 0으로 돌아온다.
- ▷ 퀘스트 기한이 초과하거나, NPC의 호감도를 연속 3번 하락시키면 명성이 2만큼 깎인다.
- ▷ 명성 수치가 -15이하가 되면 '별의 낙인이 걸린' 아이템을 얻는다.
- ▷ 명성 수치가 -15이하가 되면 위의 아이템을 얻고 수치가 0으로 돌아온다.

게임 내 변경 사항

- ▷ '별의 징표가 담긴' 아이템이 많을수록 NPC가 판매하는 물건의 가격이 하락한다.
- ▷ '별의 낙인이 걸린' 아이템이 많을수록 NPC가 판매하는 물건의 가격이 상승한다.
- ▷ NPC 물건의 판매 가격 = 원가 - 원가 X ('별의 징표 개수' - '별의 낙인 개수') / 8
- ▷ '별의 징표가 담긴' 아이템이 많을수록 NPC의 호감도 상승이 쉬워진다.
- ▷ '별의 낙인이 걸린' 아이템이 많을수록 NPC의 호감도 상승이 어려워진다.
- ▷ 일반 대화 호감도 상승량 = 20 + 2 X ('별의 징표 개수' - '별의 낙인 개수')
- ▷ 특수 대화 호감도 상승량 = 100 + 5 X ('별의 징표 개수' - '별의 낙인 개수')

게임 내 UI



출처: <https://cafe.daum.net/GuardianTales/ARs6/141195>

현재 명성 수치를 알려주며, 명성 수치가 (-)일 수록 어둠에 가깝고, (+)일 수록 빛에 가까운 UI 형식으로 존재한다.
게이지의 빛과 어둠의 경계선은 수치로 0을 표시한다.

퀘스트에 대한 종류와 각 종류에 대한 난이도 설계, 퀘스트 내역 등을 제작하였습니다.

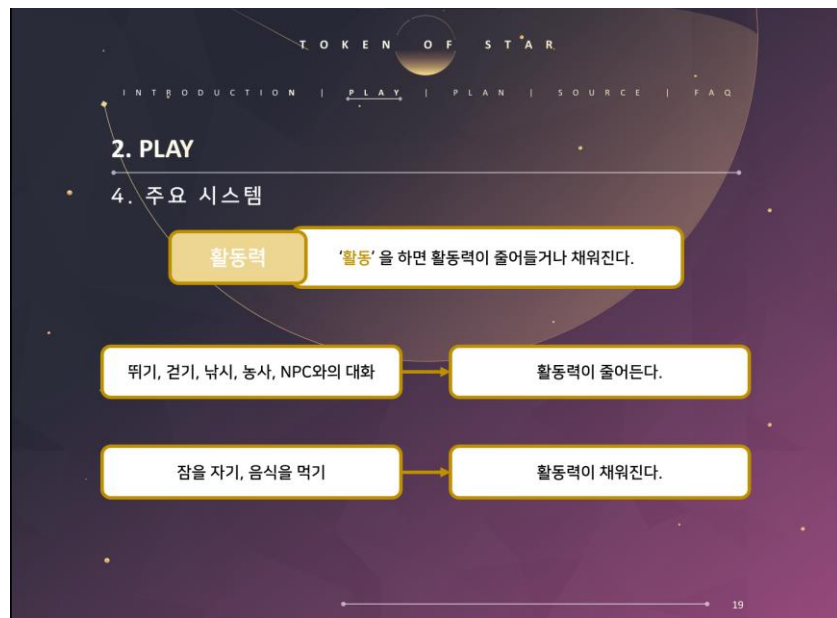
호감도의 경우 NPC에게 선물과 대화를 통해 호감도를 올릴 수 있고, 특정 호감도 수치에 달성할 경우 아이템을 받거나 특수한 퀘스트를 받을 수 있도록 하였습니다.

명성의 경우 명성이 낮을 때와 높을 때의 실점과 이점을 설계하였고 엔딩과도 이어지게끔 하였습니다.

03. 개인 프로젝트

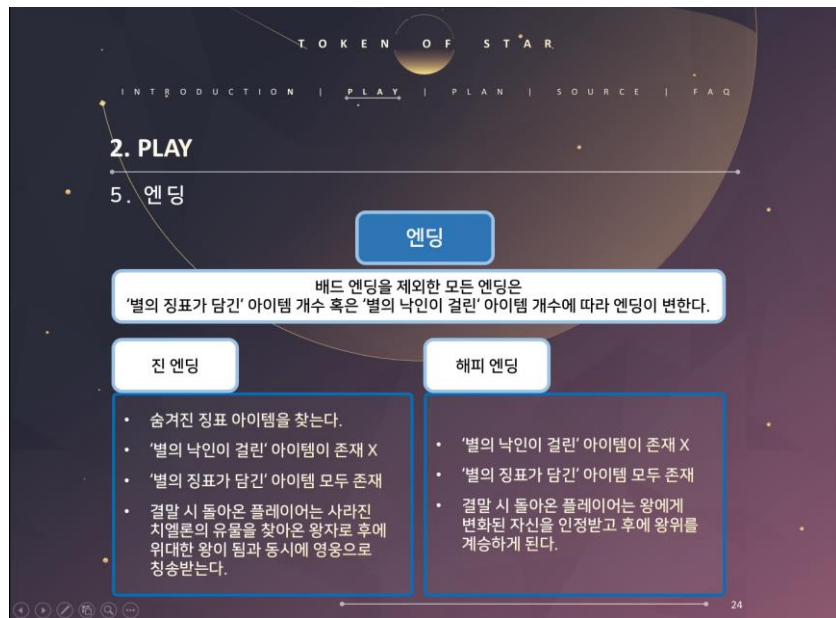


[그림 44, 45] Token of Star 활동력에 대한 설명

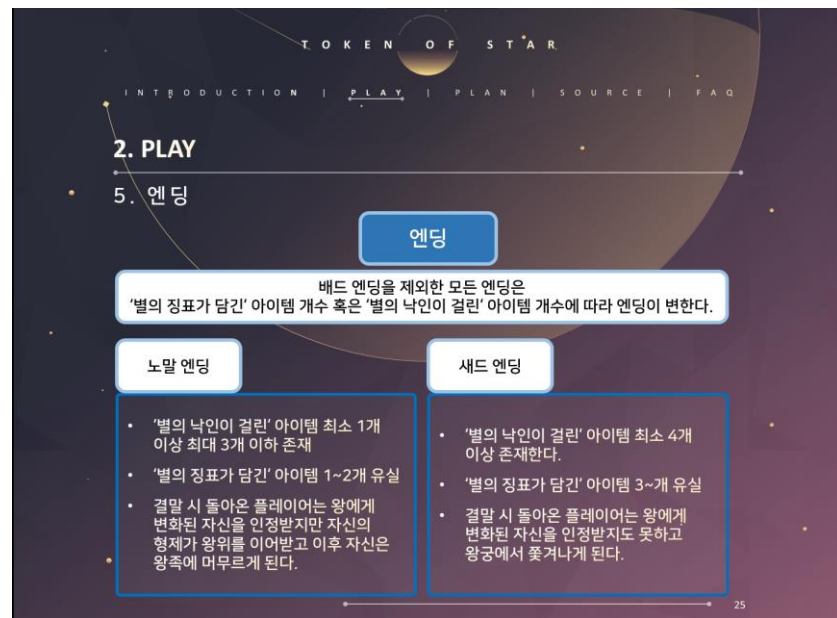


스토리에 따라 플레이어가 생활을 유지하기 위해서는 활동력이라는 시스템을 이용해야 합니다. 또한 아무것도 하지않고 게임 시간이 일정 시간 지나면 자동으로 엔딩을 보게됩니다.

03. 개인 프로젝트



[그림 46, 47] Token of Star 아이템에 따른 엔딩



플레이하면서 얻은 명성과 아이템을 통해 마지막 스토리 진행 때의 엔딩이 결정됩니다. 위의 엔딩과 배드 엔딩을 합쳐 총 5개의 엔딩을 설정하였습니다.

03. 개인 프로젝트

The image shows a presentation slide for the 'Token of Star' project. At the top, the title 'TOKEN OF STAR.' is displayed in a stylized font. Below it, a navigation bar includes 'INTRODUCTION | PLAY | PLAN | SOURCE | FAQ', with 'PLAN' highlighted. The main heading is '3. PLAN (3차)'. Below this is a table with 5 columns: '차수' (Round), '제출일' (Submission Date), '내용' (Content), '비고' (Remarks), and '변경 사항' (Changes). The table lists tasks for four rounds. Round 1 tasks include flow chart creation, screen composition, and story addition. Round 2 tasks include activity creation, initial storyboard and time planning, and quest main content composition. Round 3 tasks include quest main content detailization, mood and naming system creation, and item-based story addition. Round 4 tasks include UI/UX and skill scope/story detailization. The slide number '29' is at the bottom right.

차수	제출일	내용	비고	변경 사항
1차	22.10.19	1. 플로우 차트 작성	게임 시스템 도식화	
		2. 화면 구성	단순 UX/UI	2차시에 진행
		3. 스토리 추가		
2차	22.11.09	1. 활동	엑셀로 작성	
		2. 기초 스테이터스 및 시간	현실 시간과의 배율 및 게임 시작 시 스테이터스	
		3. 퀘스트 큰 목차 구성	자세한 내용은 3차시	엑셀로 작성
3차	22.11.16	1. 퀘스트 목차 별 내용 구체화	퀘스트 아이템에 따른 스토리 추가	엑셀로 작성
		2. 호감도, 명성 시스템 작성	호감도, 명성과 재화와의 관계 구축	엑셀로 작성
4차	22.11.16	1. UI/UX	시각화 하기 위함	엑셀로 작성
		2. 스킬 범위 표시, 세부 스토리	시각화 하기 위함	엑셀로 작성

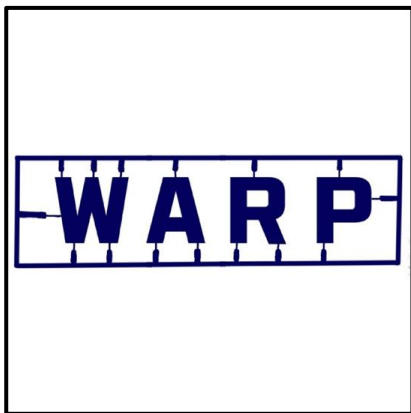
[그림 48] Token of Star에 대한 계획

해당 기획서는 위의 항목들이 완료가 된 기획서입니다.
그러나 세부 스토리와 퀘스트 내역을 다듬어서 완전히 완료할 예정입니다.

04

교내 활동

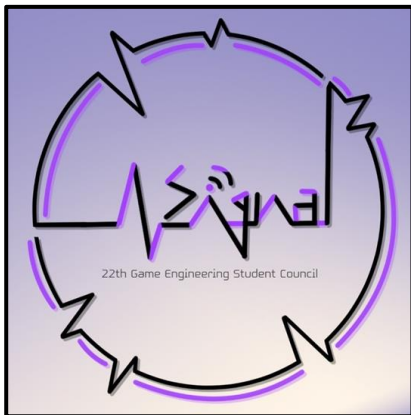
04. 교내 활동



[그림 49] WARP 소모임 아이콘

학과 공부 소모임

- 학과 공부 소모임 "WARP"
- 22년 부터 23년 현재까지 멘토
- C/C++ 비롯한 1학년 전공 과목 담당



[그림 50] 23년 게임공학과 학생회 아이콘

학과 운영기획부

- 23년 게임공학과 운영기획부 부장
- 18년 게임공학과 운영기획부 부원
- MT 및 축제 등의 큰 행사 운영 및 기획 담당



감사합니다.