

Deus Ex Machina

Dev Docs

인하대학교 미래인재개발원

AR 인공지능 게임 프로그램 개발과정

김세현, 고형주, 이수현

프로젝트 기간

2022.09.20 – 2022.12.02

목차

게임 소개.....	4
게임 장르	4
게임 내용	4
게임 시스템 설명.....	5
계정 서비스.....	5
네트워크 시스템	6
DataManager	6
상호작용 시스템	9
Casting System	15
추적 시스템.....	19
Behavior 시스템	20
Behavior Class	21
Player Controller	24
Camera Controller.....	25
Cinemachine Camera Setting	26
Post-Processing Setting.....	27
Contents	28
퇴마사 플레이어	28
인형 플레이어 캐릭터	38
오브젝트	43
Scene 소개.....	55
00 Launcher.....	55
00 Launcher_LOGIN	57

02_MainLobbyScene	58
03_SelectCharacterScene.....	65
03_RoomScene	66
MainGameScene	68
97>LoadingScene	70
98>LoadingNetworkScene	71
99_GameResultScene	72
Asset List	74
Class Diagram	76

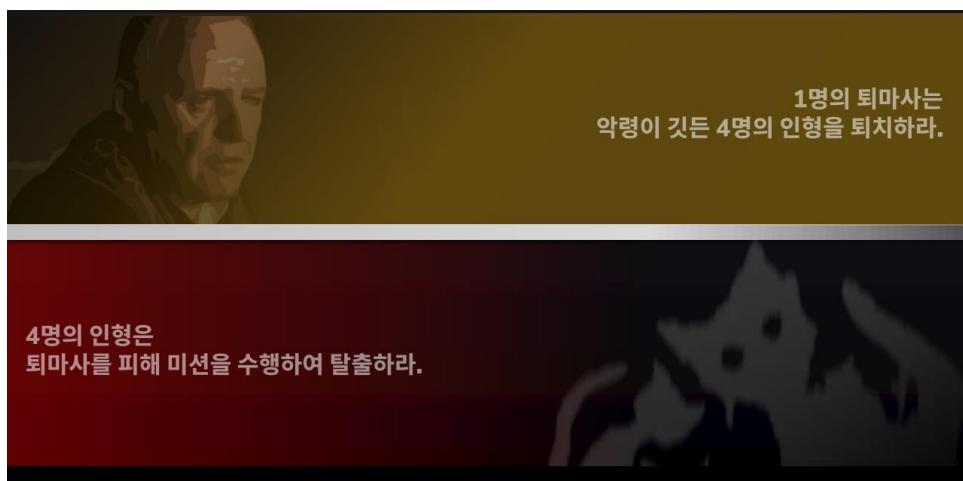
게임 소개

게임 장르

비대칭 서바이벌 PVP 게임



게임 내용

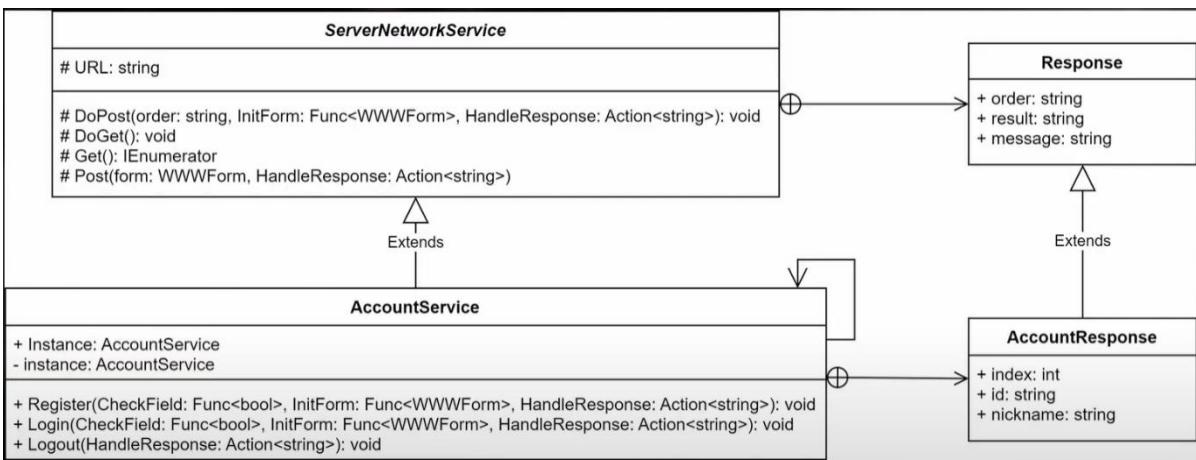


플레이어는 퇴마사와 인형 중 하나의 진영을 선택하여 플레이를 진행합니다.

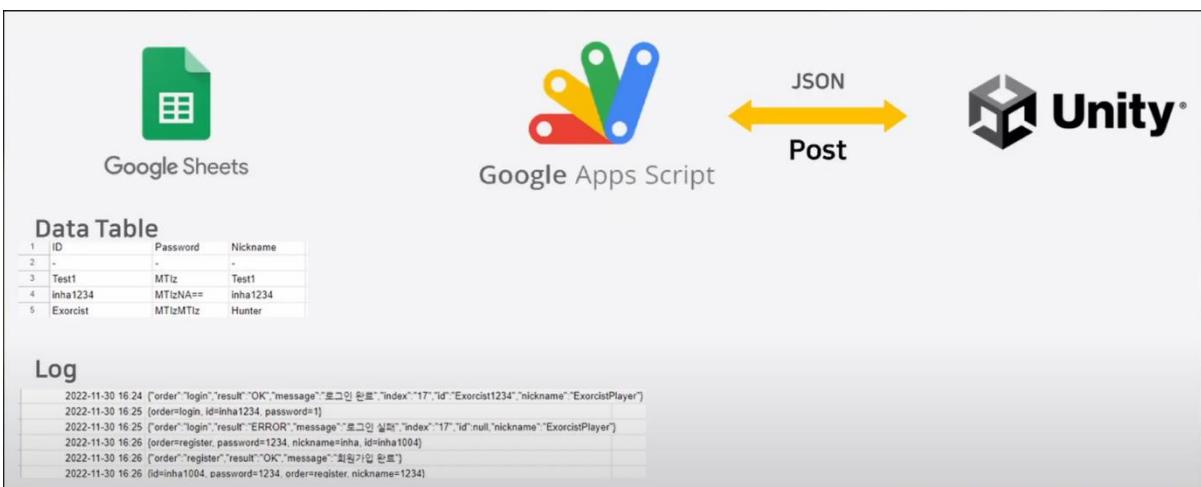
퇴마사는 술래잡기의 술래와 같은 역할로 맵에 존재하는 인형 플레이어를 모두 제거하는 것이 목표이며, 인형은 퇴마사를 피해 탈출하는 것이 게임의 목표입니다.

게임 시스템 설명

계정 서비스



계정 로그인 및 회원가입을 위한 시스템입니다.

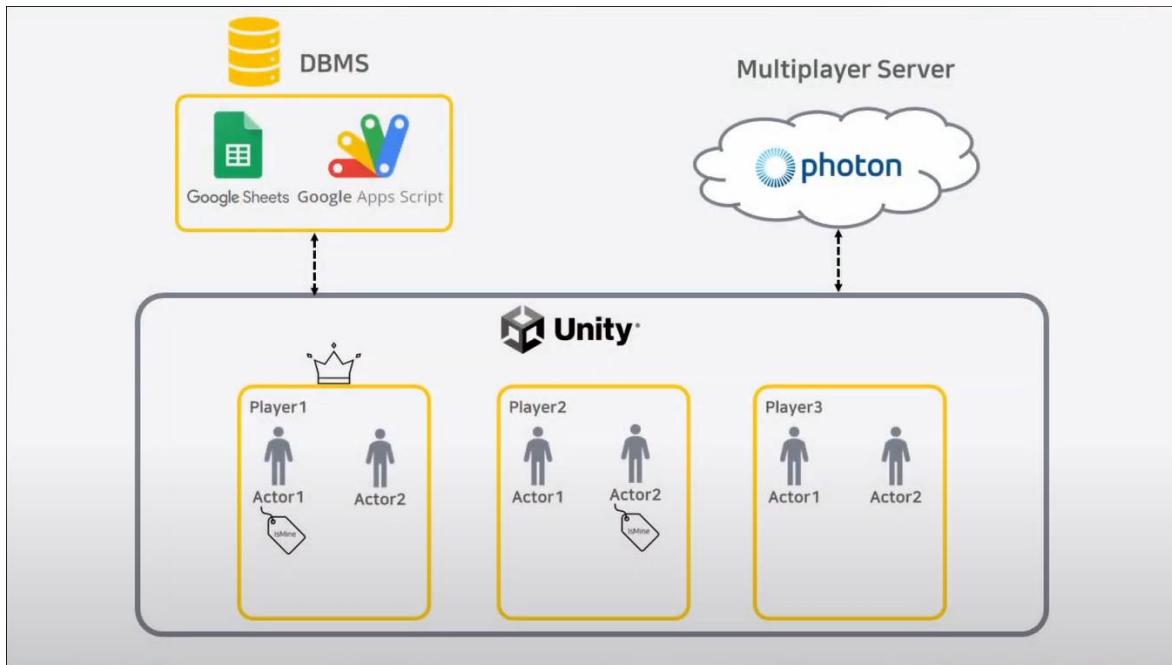


DBMS 를 구현하기 위해 Google Sheet 와 Google Apps Script 를 사용했습니다.

DBMS 를 사용 할 수 있는 환경이 아니어서, DBMS 를 가상으로 구현하였습니다.

Google Sheet 를 DB 테이블로 두고, Google Apps Script 를 통해 Unity 와 데이터를 JSON 으로 직렬화하여 POST 로 주고받게 되어있습니다.

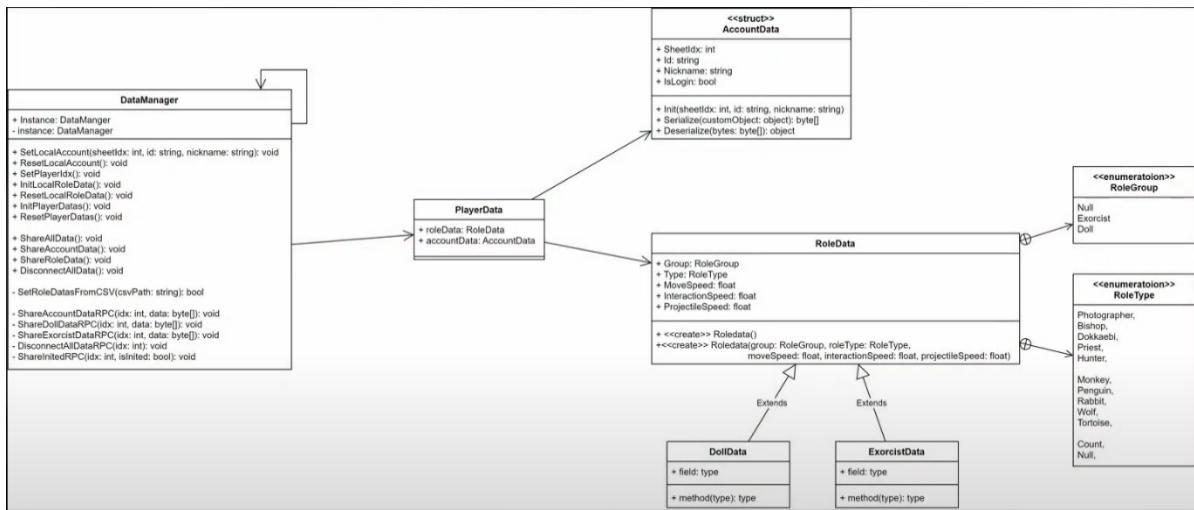
네트워크 시스템



멀티플레이어 환경을 위해 Photon PUN2 API를 사용합니다.

따라서 저희 게임에서는 멀티플레이어를 위한 Photon 클라우드 서버와, DB를 위한 서버 두 가지를 사용하고 있습니다.

DataManager



DataManager는 플레이어의 계정 정보와 게임 정보를 관리하기 위해 제작한 클래스입니다.

DataManager는 CSV 파일로 저장되어있는 캐릭터 정보를 쓴 초기에 불러와서 메모리에 적재 하며, 해당 데이터를 토대로 런타임에 수정 가능한 플레이어 정보를 생성합니다.

DataManager는 클라이언트의 플레이어 정보와, 다른 플레이어의 정보를 List로 가지고 있습니다.

이는 게임에서 플레이어가 다른 플레이어의 데이터를 참고하기 위함입니다.

또한 List에서 자신의 데이터는 오직 해당 클라이언트의 플레이어만 수정이 가능하며, 자신의 클라이언트가 아닌 데이터는 수정이 불가능하도록 제작하였습니다.

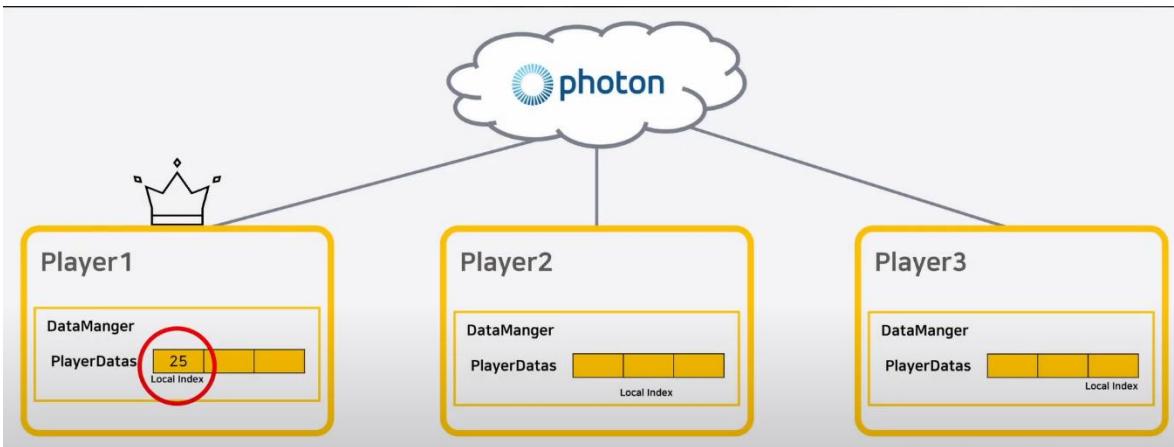
이를 통해 다른 플레이어의 조작을 통해 데이터가 손상되는 것을 방지 하였습니다.



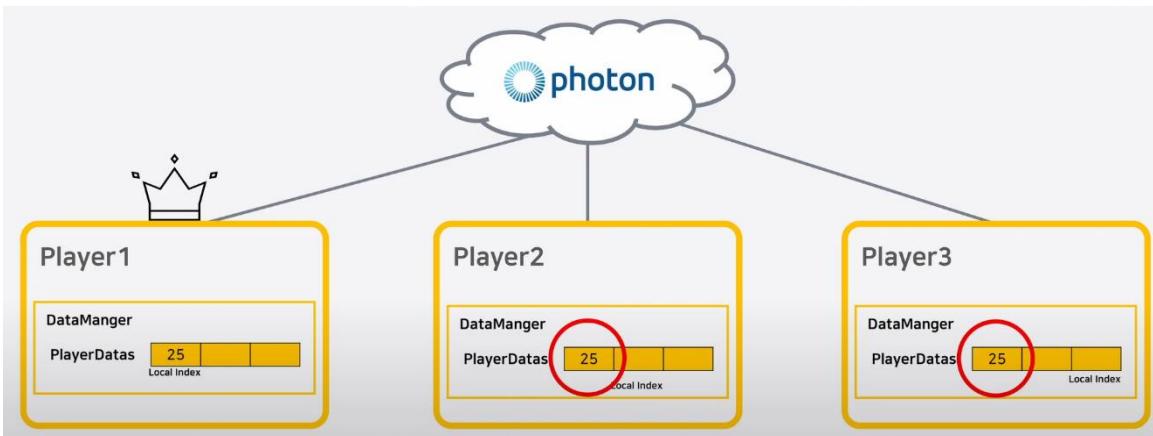
DataManager는 데이터 공유를 쉽게 하기 위하여 Photon PUN2 API를 Façade 패턴으로 래핑합니다.

이를 통해 다른 클래스에서 플레이어 정보를 변경하는 경우, 별도의 API 사용 없이 DataManager의 동기화 함수를 사용하도록 제작하였습니다.

이러한 동기화 인터페이스 함수는 Photon PUN2 API의 RPC 함수를 대신 처리하여 자신의 데이터를 다른 클라이언트와 동기화 합니다.



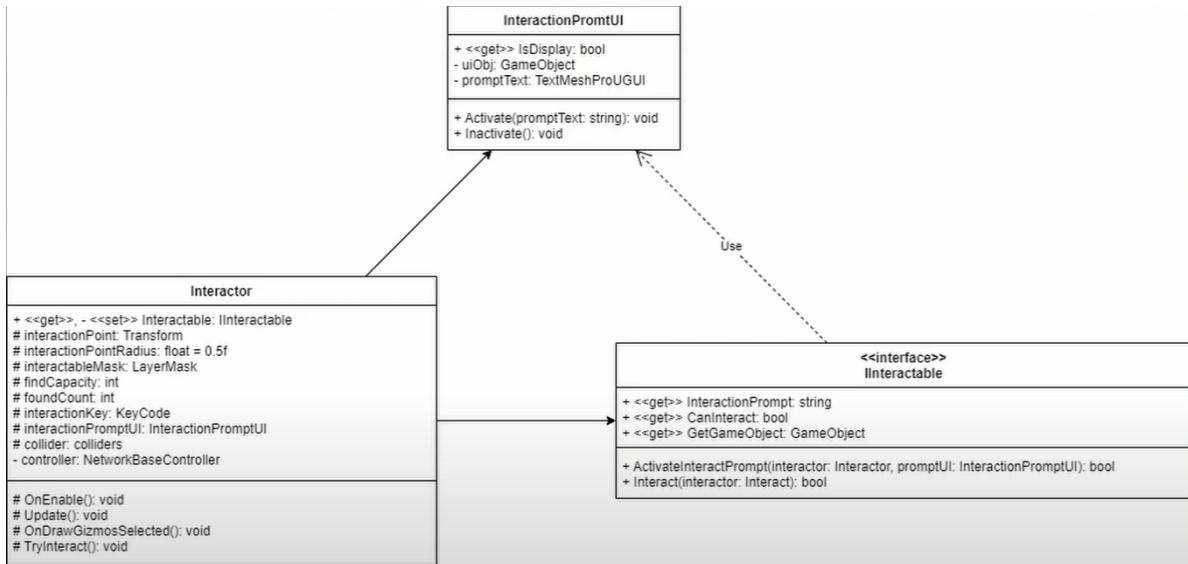
예를 들어 Player1 이 자신의 데이터를 변경한 후 공유를 요청하면,



DataManager 는 Photon RPC 함수를 통해 다른 클라이언트에 자신의 데이터를 동기화 시킵니다.

위 동기화 과정에서 각 데이터들은 기본 자료형이 아닌 경우 모두 byte 로 Serialize 및 Deserialize 하여 데이터를 공유합니다.

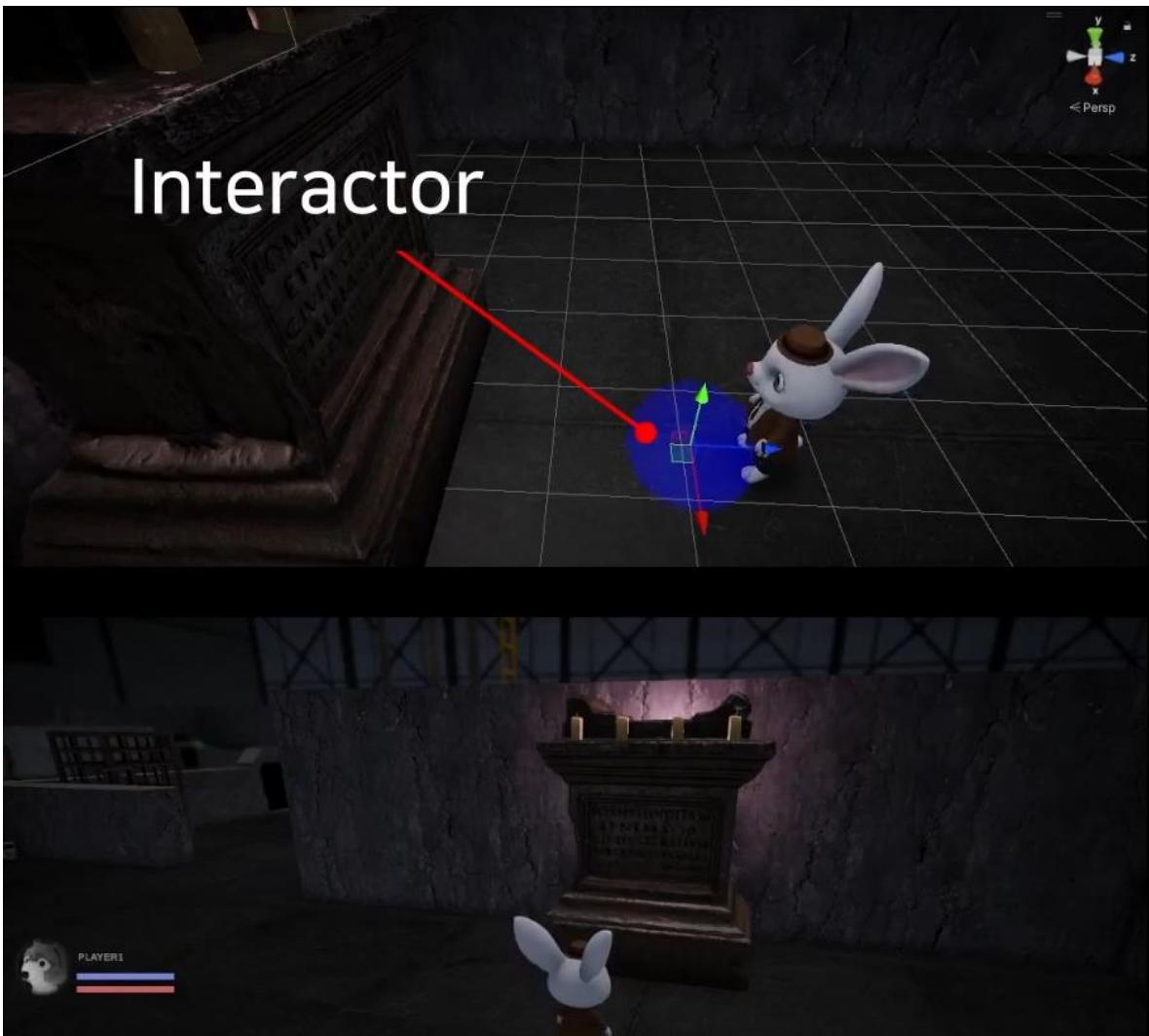
상호작용 시스템



상호작용 시스템은 플레이어와 다른 오브젝트 사이에서 상호작용을 하기 위한 시스템입니다.

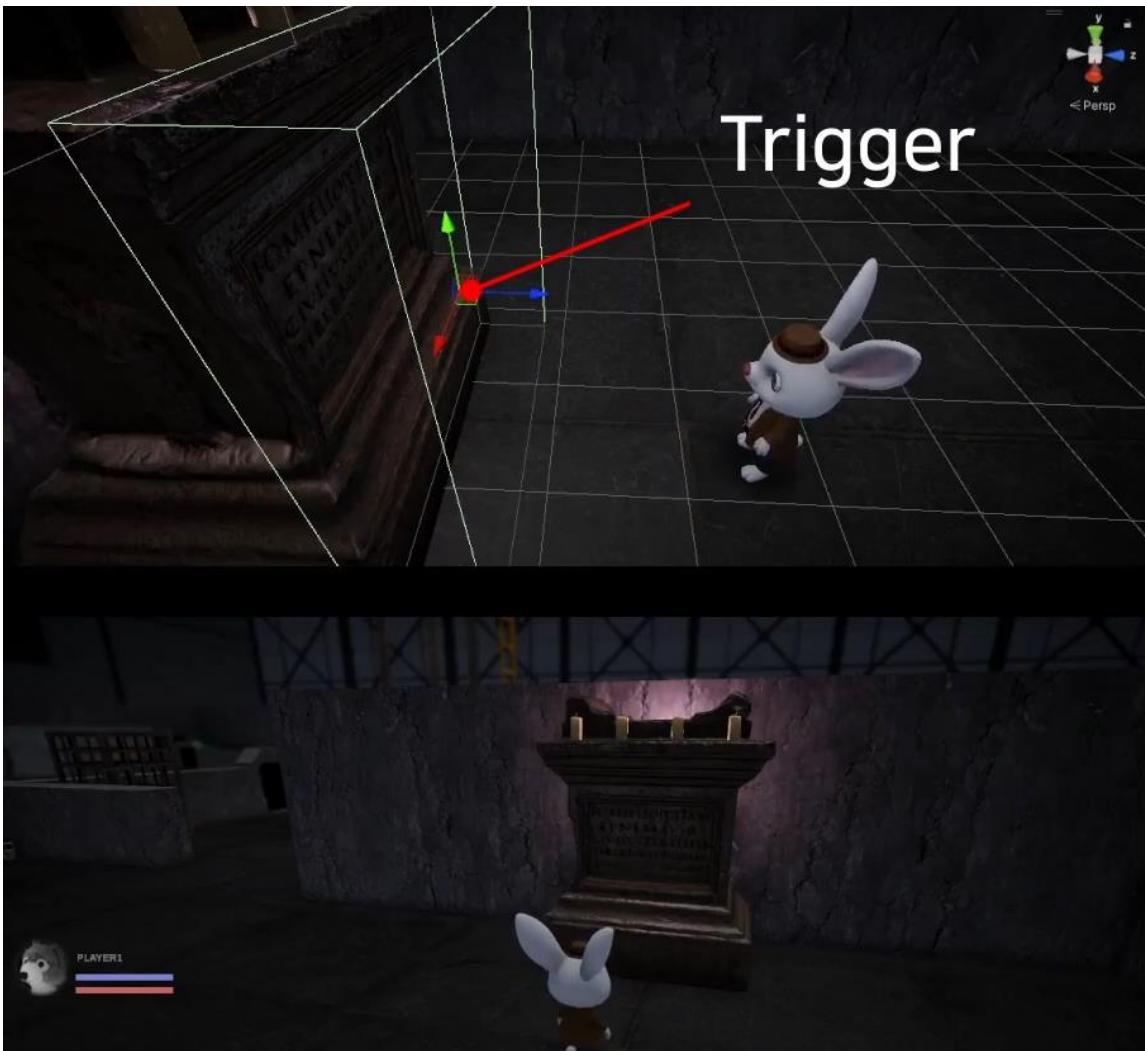
`Interactor`라는 상호작용이 가능한지 판단하는 Component 클래스가 있으며, `IInteractable` 인터페이스를 가진 클래스를 탐색합니다.

만약 `Interactor`가 `IInteractable`을 구현하는 클래스를 찾은 경우, `TryInteract()`를 실행하며, 상호작용이 가능한 경우 `InteractionPromptUI`에서 상호작용이 가능하다는 메시지를 출력하도록 합니다.



예를 들어 플레이어는 Interactor 를 자신의 앞에 가지고 있습니다.

0| Interactor 에는 중심점과 반지름을 가지고, 주변의 IInteractable 을 탐색합니다.



위 오브젝트는 정면에서 상호작용이 가능하도록 기획된 오브젝트로, 트리거는 위와 같이 설정되어 있습니다.



Interactor 가 오브젝트의 트리거와 만난 후, 상호작용이 가능한 상태임을 감지하면, UI 에는 상호작용이 가능한 상태를 알리는 메세지가 출력되며, 오브젝트에는 외곽선을 그립니다.

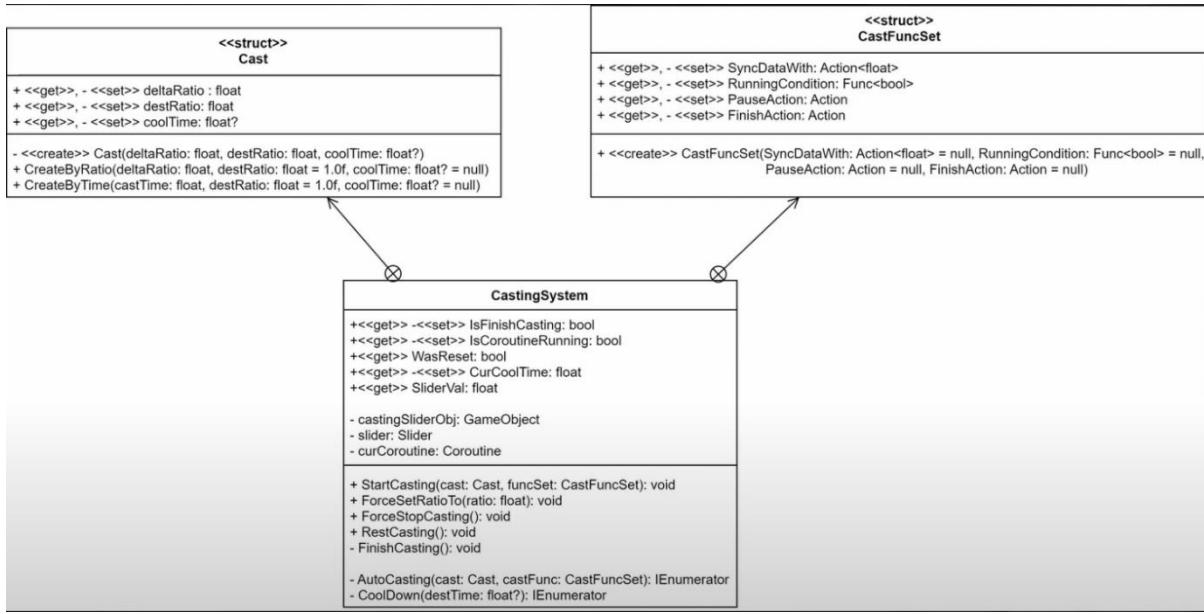


상호작용이 불가능한 상태가 되면 다시 외곽선을 없애고, 출력 메세지를 띄우지 않습니다.



상호작용이 가능한 상태에서 플레이어가 상호작용 키를 누르는 경우, `IInteractable` 을 구현하는 클래스의 해당 상호작용 행동을 수행합니다.

Casting System



Casting System 은 게이지를 총전하기 위한 시스템입니다. 저희 게임에서는 대부분의 상호작용에서 캐스팅을 하기 때문에 위와 같은 시스템을 제작하였습니다.

CastingSystem 클래스는 Casting 의 타입을 나타내는 Cast 클래스와, Casting 을 수행할 때 이루어질 함수들의 조합인 CastFuncSet 구조체로 이루어지며, 이 둘을 이용한 빌더 패턴으로 제작하였습니다.



Casting System에는 각각 두 가지의 종류와 저장 방식이 존재합니다.

Casting Type

Auto

한 번 키를 누르면 끝까지 자동으로 작동하여, 게이지가 다 찰 때까지 캐스팅을 하는 타입입니다.



위 사진에서 캐릭터는 문과 상호작용하는데, 이 때 상호작용 키를 누르면 게이지가 다 찰 때까지 자동으로 게이지가 차도록 되어있습니다.

위 상황에서는 상호작용 키를 누른 후 땐 상태입니다.

Manual

상호작용 키를 누를 때만 작동하여, 해당 상황에서만 게이지가 차는 타입입니다.



위 사진에서 캐릭터는 제단과 상호작용 하는데, 이 때 상호작용 키를 계속 누르고 있어야지만 게이지가 차도록 되어있습니다. 만약 키를 때는 경우 상호작용이 종료됩니다.

Gauge Storage Method

Volatile

캐스팅 도중 캐스팅을 취소하는 경우 캐스팅 했던 게이지가 초기화 되는 저장방식 입니다.

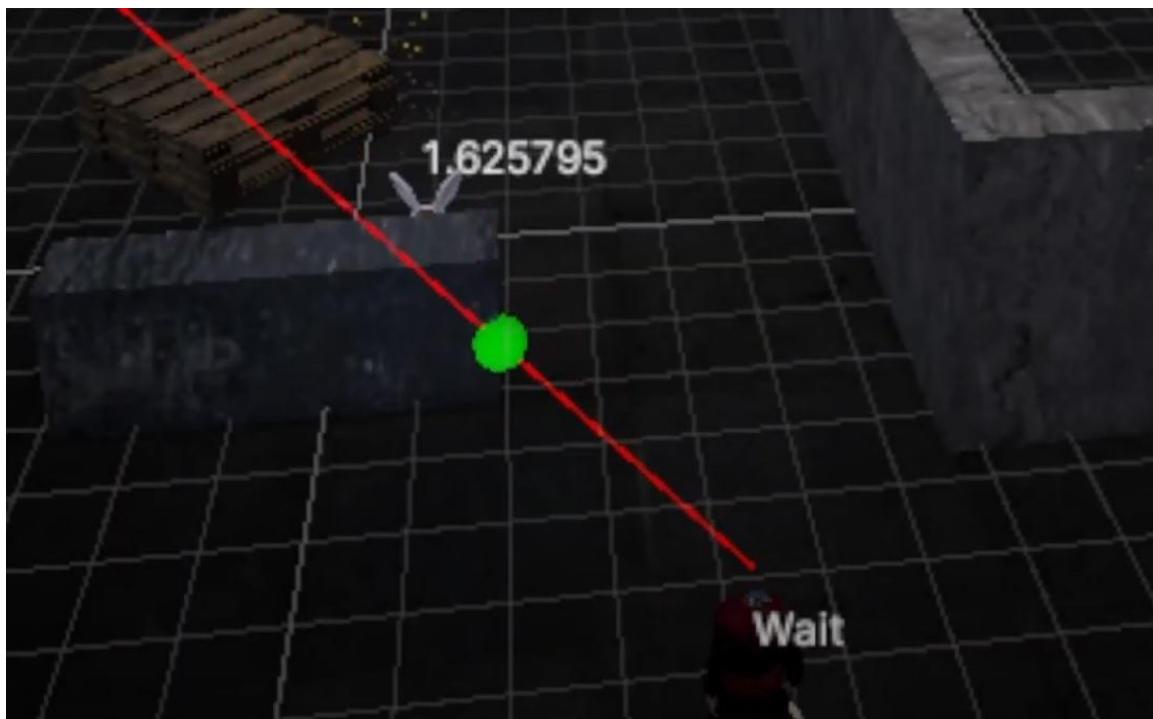


플레이어가 비상탈출 제단과 상호작용 하는 모습입니다. 만약 위 상황에서 상호작용을 취소하는 경우, 게이지는 모두 초기화 되고 다시 처음부터 캐스팅을 해야 합니다.

Non-Volatile

캐스팅 도중 캐스팅을 취소해도 캐스팅했던 게이지가 저장되어 있는 저장 방식입니다.

추격 시스템



추격 시스템은 퇴마사 캐릭터가 인형 캐릭터를 추격하는 상황을 위해 만든 시스템입니다.

추격 시스템은 퇴마사의 상태를 가지고 있으며, Wait, Chase, CoolDown 으로 총 3 가지의 상태를 가지고 있습니다.

- Wait: 아무 일도 없는 상태
- Chase: 추격 상태
- CoolDown: 추격 상태가 종료되고 주는 예비 시간

퇴마사는 인형 감지 거리를 가지고 있으며, 만약 감지 거리 내에 인형이 들어오는 경우, 인형을 향해 Raycast 합니다. 이 때 Ray 사이에 장애물이 있으면, 퇴마사가 인형을 발견하지 못한 것으로 판단합니다.

하지만 Ray 가 장애물 없이 상대방과 만나고, 퇴마사 플레이어의 스크린 안에 인형이 보인다면, 퇴마사는 인형을 발견한 상태가 됩니다.

퇴마사가 인형을 발견하면, 인형의 추격 수치가 상승하며, 추격 수치가 지정된 숫자를 넘어가는 경우, 퇴마사는 추적 상태로 변환됩니다.

추적 상태가 되면 퇴마사와 인형 모두 추격 상태 배경음악을 출력됩니다.

퇴마사가 인형을 발견하지 못하면, 인형의 추격 수치는 줄어듭니다. 추격 수치가 지정된 숫자보다 낮은 경우 퇴마사의 추적 상태가 풀립니다.

추적 상태가 풀린 경우 퇴마사의 상태는 지정된 시간 동안 CoolDown 으로 변경되며, 해당 상태에서는 인형을 발견해도 추격 수치가 증가하지 않습니다.

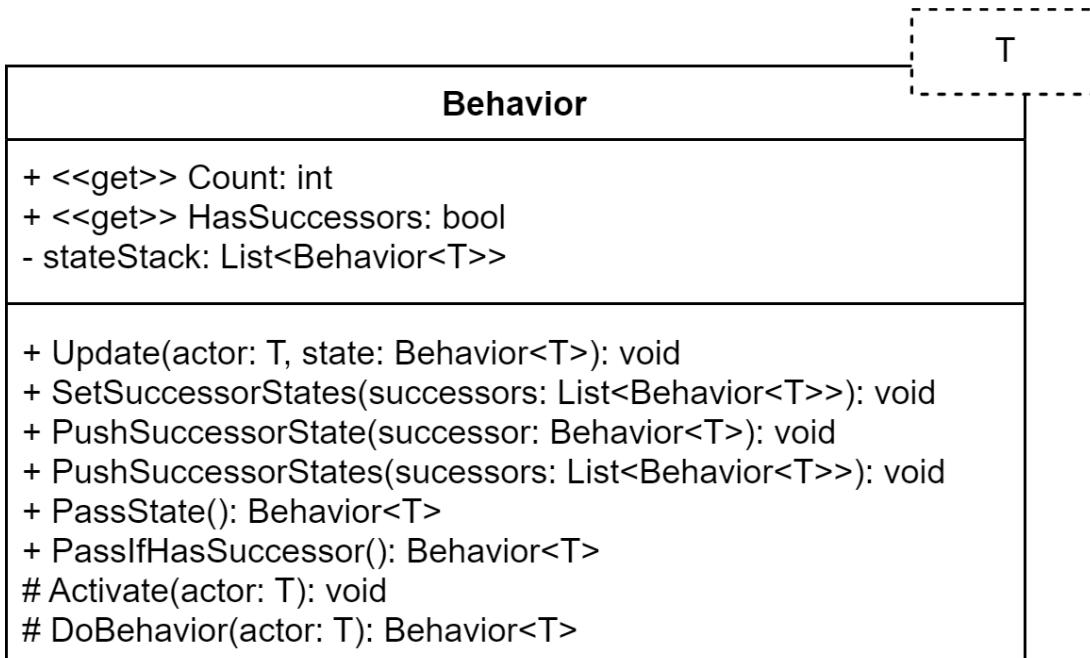
지정된 시간이 모두 지나면 퇴마사의 상태는 Wait 으로 변경되며, 다시 추격 시스템이 작동할 수 있는 상태가 됩니다.

Behavior 시스템

게임에서 캐릭터의 행동 상태와 스킬은 모두 Behavior 클래스를 이용해서 제작되었습니다.

Behavior 클래스는 State 를 List 로 가지고 있으며, 이를 이용하여 Stack-based FSM 을 구현하고 있습니다.

Behavior Class



정의

네임스페이스: KSH_Lib

캐릭터 클래스 T의 행동 상태를 리스트로 가지고 있는 클래스입니다.

PDA (Push Down Automata) 또는 SFSM(Stack Finite State Machine)의 형식으로 상태를 관리합니다.

여러 개의 행동 상태를 행동 리스트에 담아 스택으로 사용하며, 리스트의 끝부터 꺼내어 차례로 행동을 수행합니다.

행동 리스트에는 행동 상태 하나를 추가할 수 있으며, 여러 개의 행동 상태 리스트 또한 추가가 가능합니다.

형식 매개 변수

T

Behavior를 가질 클래스 형식입니다.

Fields

List<<Behavior T>> stateStack

행동 상태들을 저장하는 리스트입니다. 가장 마지막 원소부터 꺼내어 사용합니다.

Count

List<<Behavior T>> stateStack 의 수를 가져오는 변수입니다.

Methods

public virtual void Update(in T actor, ref Behavior<T> state)

Behavior 의 상태를 업데이트하는 가상 함수입니다.

public virtual void Activate(in T actor)

해당 상태에 처음 들어올 때 한 번만 호출되는 가상 함수입니다.

public virtual Behavior<T> DoBehavior(in T actor)

해당 상태에서 반복적으로 수행되는 함수입니다.

기본적으로 반복 수행을 할 때는 null 값을 반환하여 현재 상태를 유지할 수 있으며, 상태 변경이 필요한 경우 해당 상태를 반환하여 상태를 전환할 수 있습니다.

public void SetSuccessorStates(in List<Behavior<T>> successors)

stateStack 을 새로운 상태들로 바꾸는 함수입니다.

public void PushSuccessorState(in Behavior<T> behavior)

stateStack 에 새로운 상태를 추가하는 함수입니다.

public void PushSuccessorStates(in List<Behavior<T>> behaviors)

stateStack 에 새로운 상태들을 추가하는 함수입니다.

public bool HasSuccessors()

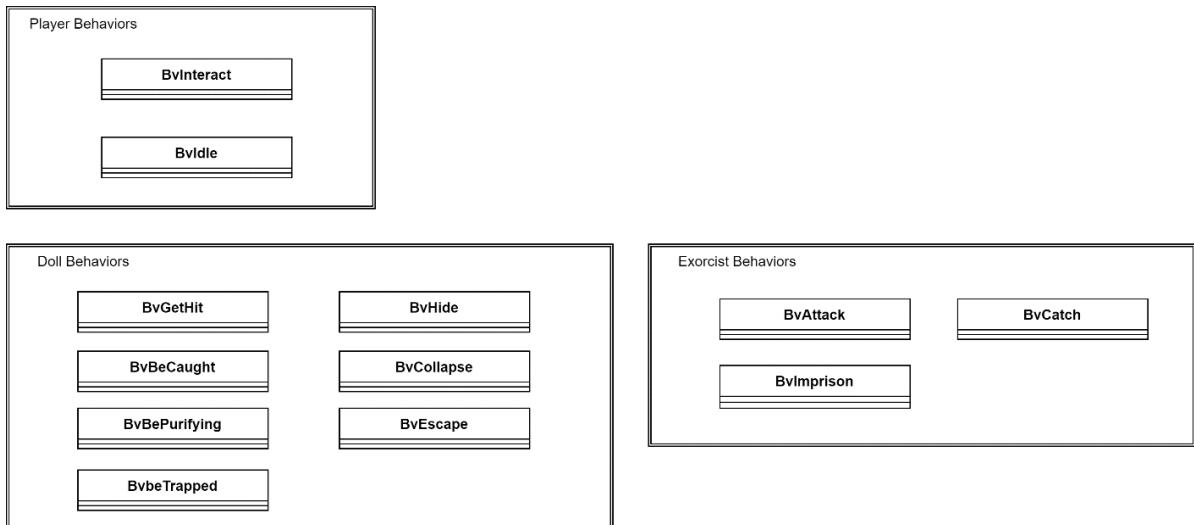
stateStack 에 상태가 있는지 체크하는 함수입니다.

public Behavior<T> PassState()

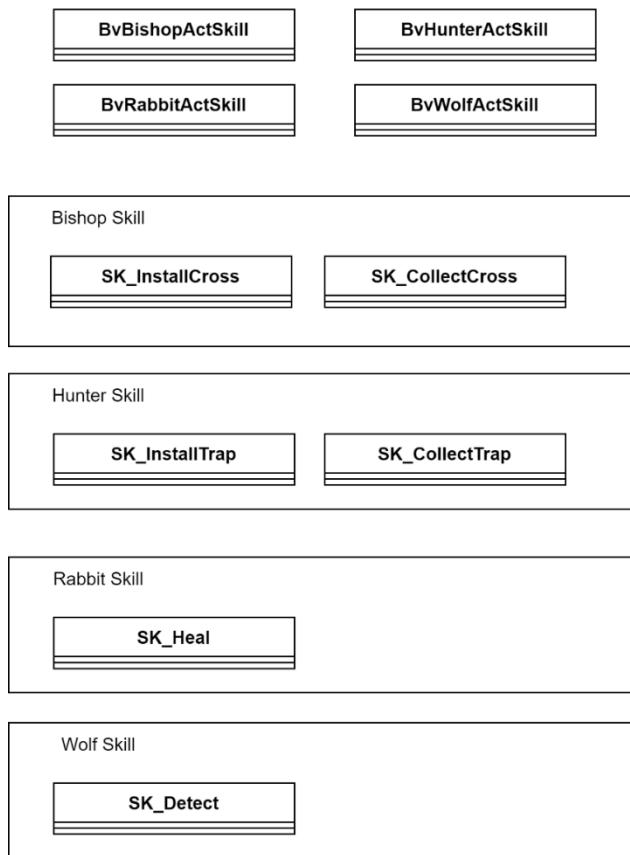
stateStack 의 가장 마지막 원소(Top)의 상태를 불러오는 함수입니다.

현재 상태는 해당 상태로 변경되며, 해당 stateStack 은 기존의 stateStack 의 마지막 원소가 사라진 상태로 변경됩니다.

Behavior 클래스를 이용해 구현된 캐릭터의 상태는 다음과 같습니다.



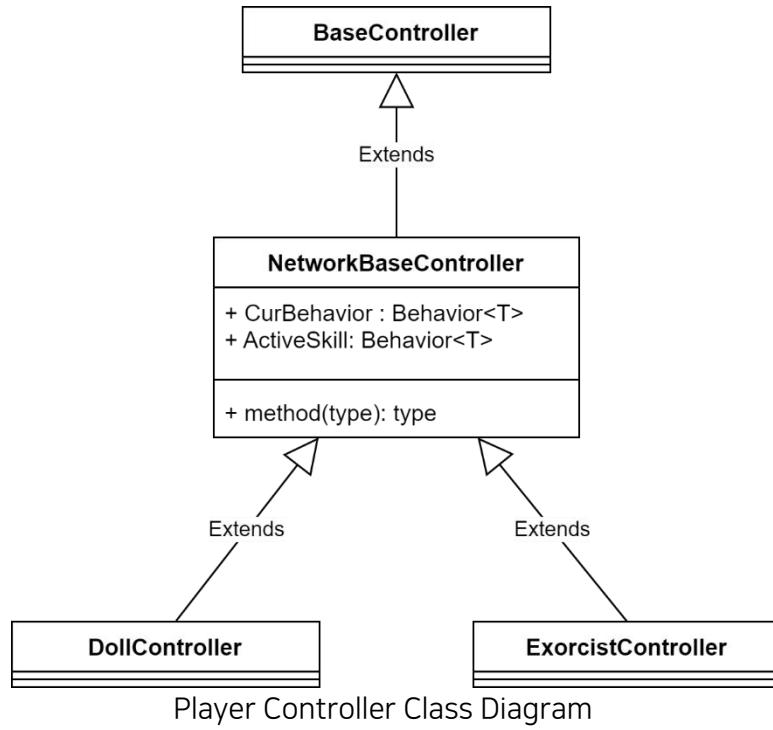
Behavior 클래스를 이용해 구현된 스킬은 다음과 같습니다.



각 캐릭터 별로 사용 가능한 스킬 행동이 있으며, 해당 스킬 행동은 여러 개의 스킬 효과로 구성되어 있습니다.

예를 들어 BvBishopActSkill 은 Sub State 로 SK_InstallCross 와 SK_CollectCross 를 가지고 있어, 이 두 행동의 조합으로 구성되어 있습니다.

Player Controller



Player Controller Class Diagram

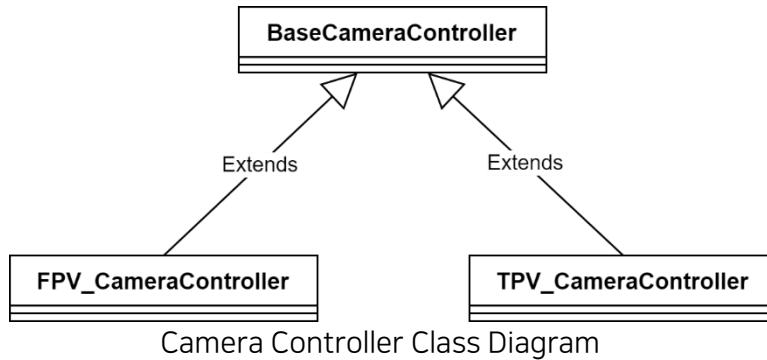
플레이어를 조작하는 컨트롤러는 위와 같은 구조를 가지고 있습니다.

BaseController는 플레이어 컨트롤러가 공통적으로 지니는 속성을 가지고 있는 클래스이며, **NetworkBaseController**는 위 클래스를 상속받는 네트워크 버전의 컨트롤러입니다.

플레이어 캐릭터는 총 두 가지의 타입으로 나누기 때문에, 플레이어 컨트롤러 또한 두 가지로 구현됩니다.

따라서 실제로 캐릭터의 조종 컴포넌트로 사용되는 클래스는 **DollController** 와 **ExorcistController** 입니다.

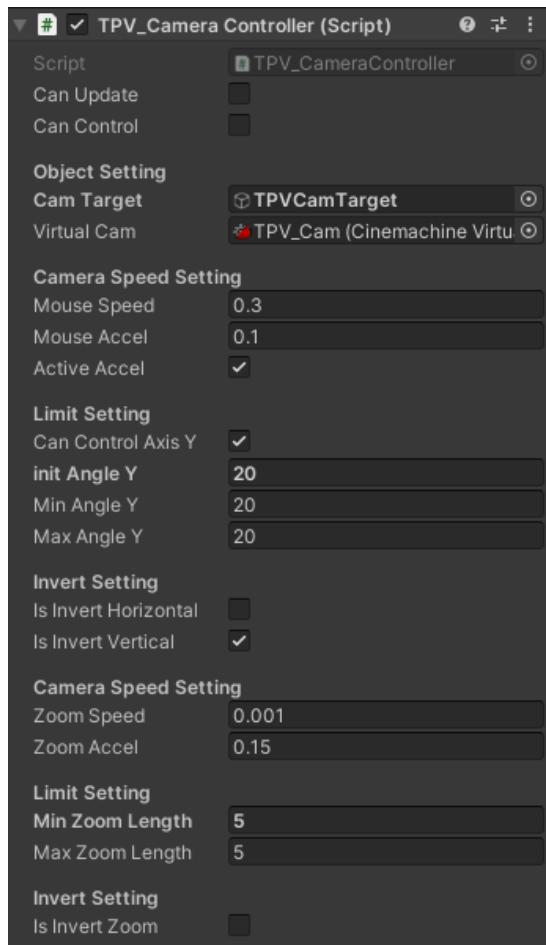
Camera Controller



CameraController 는 시네마신 카메라를 제어합니다.

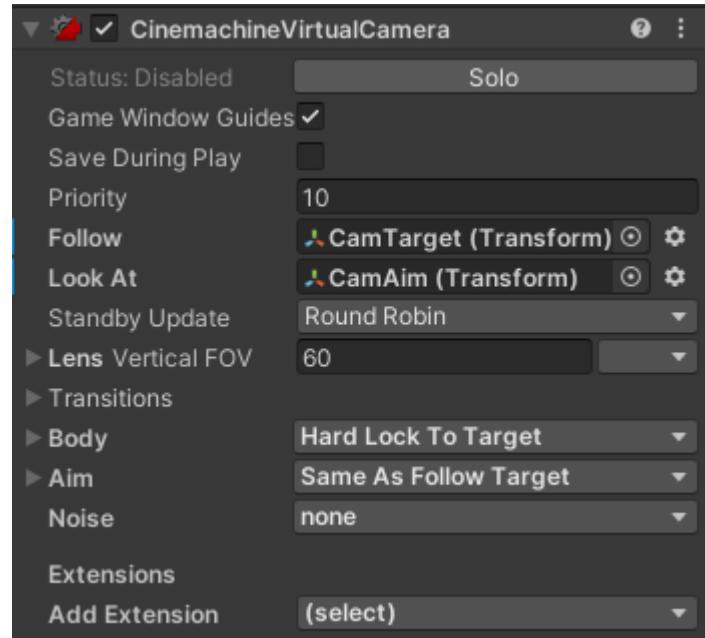
카메라는 공통적으로 가지는 속성을 지니는 BaseCameraController Class 를 상속받아 구현합니다.

1 인칭 카메라와 3 인칭 카메라는 작동 방식이 다르기 때문에 구분되어 사용됩니다.

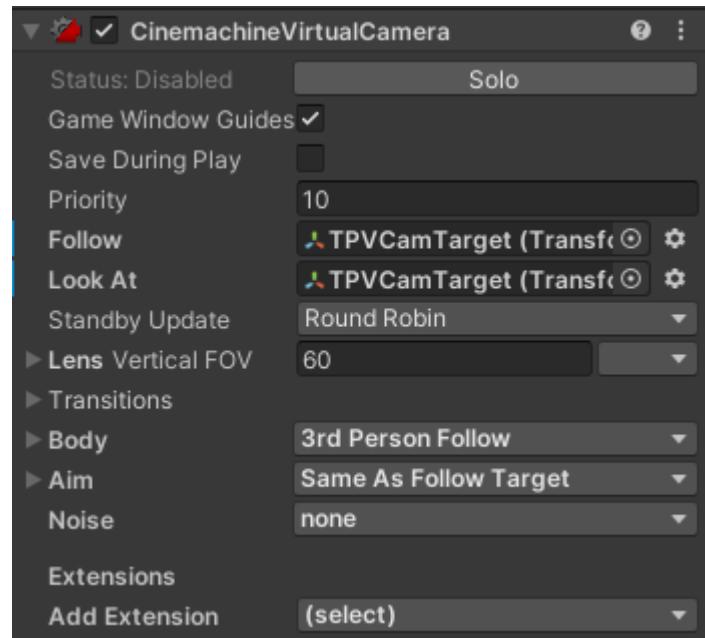


TPV_CameraController Component

Cinemachine Camera Setting

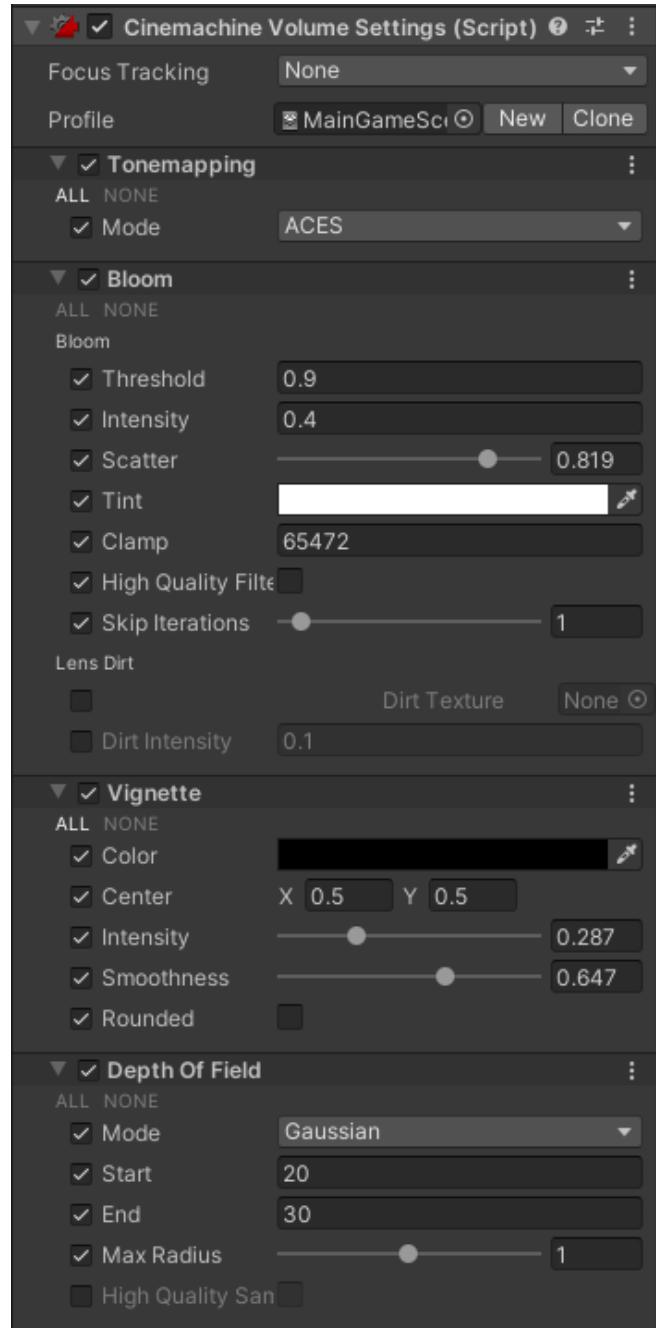


1인칭 카메라 시네마신 설정



3인칭 카메라 시네마신 설정

Post-Processing Setting



Post-Processing 설정

색상 톤을 어둡게 변경하기 위해 톤 맵핑 모드를 ACES로 설정했습니다.

이펙트에서 빛나는 것을 강조하기 위해 Bloom 효과를 넣었으며, 화면 집중력을 위해 비네트를 넣었습니다.

또한 성능의 저하가 있지만 원경을 흐리게 하기 위해 DOF를 넣었습니다.

Contents

퇴마사 플레이어



퇴마사 플레이어는 게임 씬에서 혼자만 가지는 직업이며, Photon Server 에서는 Host 의 역할인 Master Client 로 등록되어있습니다.

퇴마사는 게임에 존재하는 다수의 인형을 공격하여 쓰러뜨린 후, 모두 정화상자에 가두어 정화 시켜야하는 목표가 있습니다.

만약 인형이 모두 탈출하는 경우 게임에서 패배하게 됩니다.

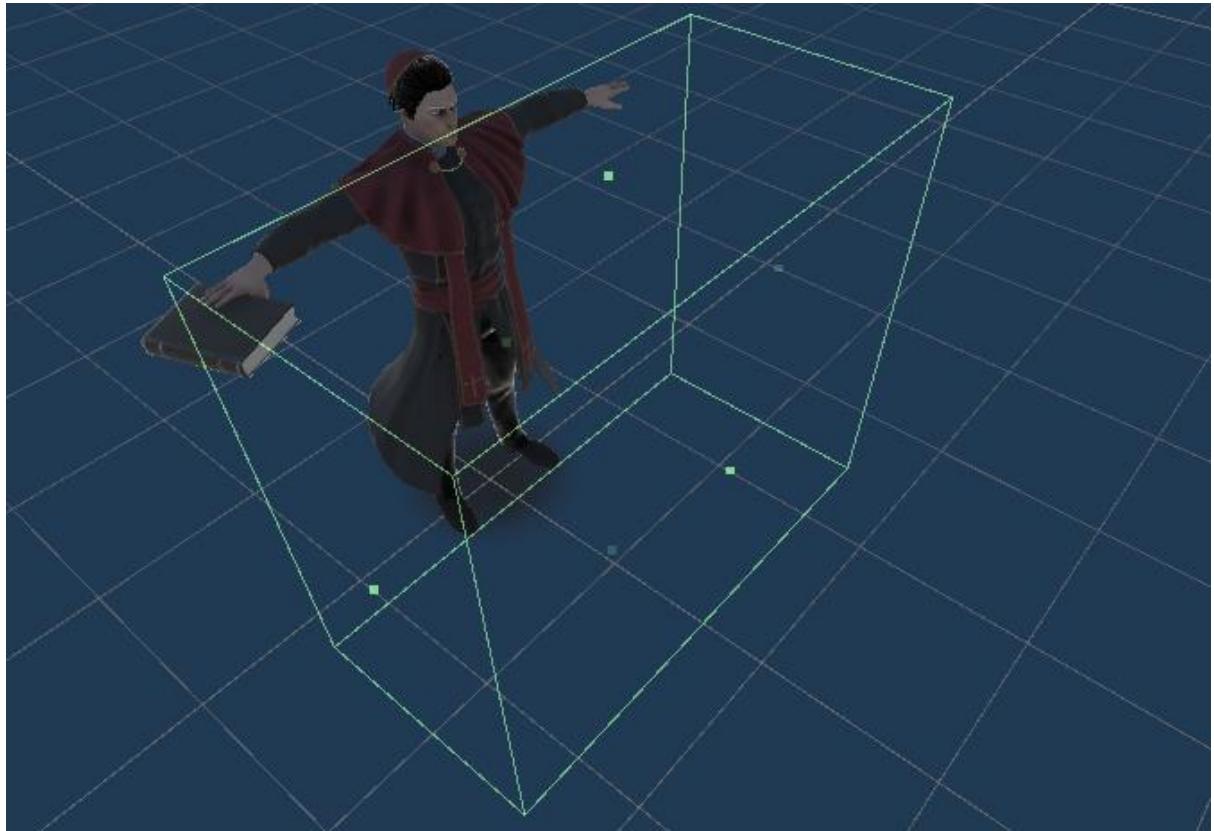
퇴마사 플레이어는 1 인칭 시점에서 게임 플레이가 진행됩니다.

퇴마사 공통 - 공격, 잡기

퇴마사는 모두 마우스 왼쪽을 눌러 BvAttack 상태로 전환하고 공격을 할 수 있습니다.

BvAttack에서 캐릭터마다 고유의 공격행동을 실시합니다.

공격 행동할 때 퇴마사 정면에 Trigger가 생성됩니다.



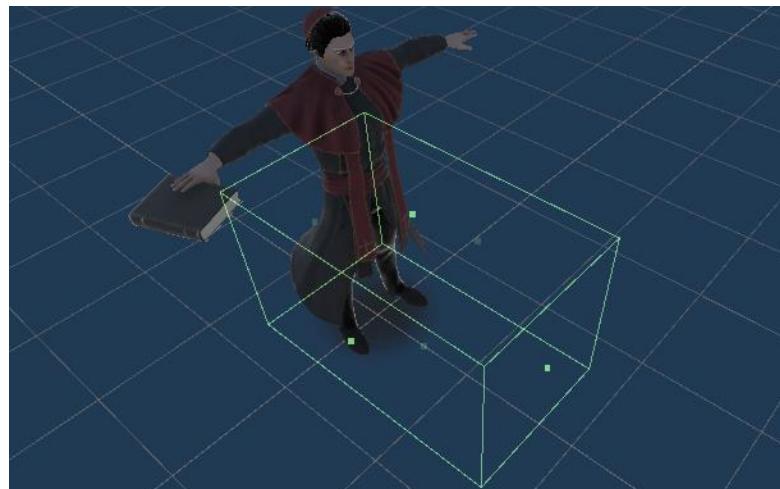
공격 모션 시간에 따라 Trigger 안에 제일 가까운 인형의 인형 HP를 낳게 합니다.

해당 인형은 BvBeGetHit 상태로 전환됩니다.

BvBeGetHit에서 인형 HP가 0 이하가 됐을 경우 인형은 BvCollapse로 상태가 전환된 후 움직이지 못합니다.

퇴마사는 또 다른 Trigger인 PickUpArea를 통해 상시 BvCollapse 상태의 인형을 검사합니다.

Trigger 범위 내에 BvCollapse의 상태를 가진 인형이 있다면 공격키를 눌렀을 때 공격행동이 아닌 해당인형을 줍는 행동을 실시합니다.



퇴마사 캐릭터가 인형을 주울 수 있는 범위

퇴마사는 BvCatch 상태로 전환됩니다.

잡힌 인형은 BvBeCaught 상태로 전환됩니다.

이 후 설명할 오브젝트의 PurificationBox 와 상호작용을 통해 BvBeCaught 인 인형을 PurificationBox 에 가두고 BvBePurifying 상태로 전환합니다.

퇴마사는 BvIdle 로 돌아갑니다.

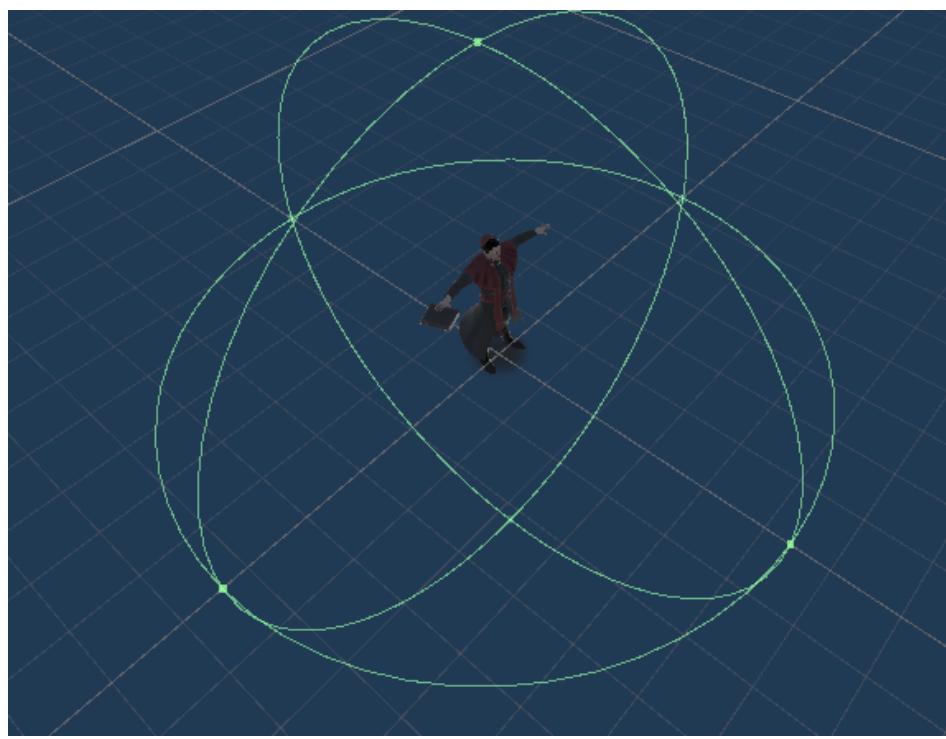
주교 캐릭터 패시브 스킬 - 악령 HP 감소

주교가 인형을 때릴 경우 인형 HP에 준 피해의 일정 비율로 악령 HP를 줄입니다.

주교 캐릭터 액티브 스킬 - 십자가 설치



설치된 십자가 모습



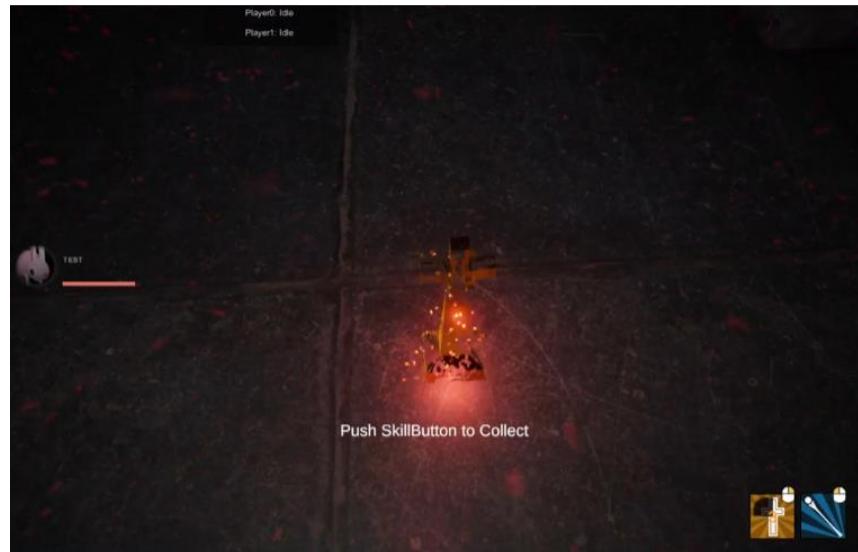
설치된 십자가 탐지 범위

스킬 범위 내에 다른 십자가가 없다면 스킬 버튼으로 설치합니다.

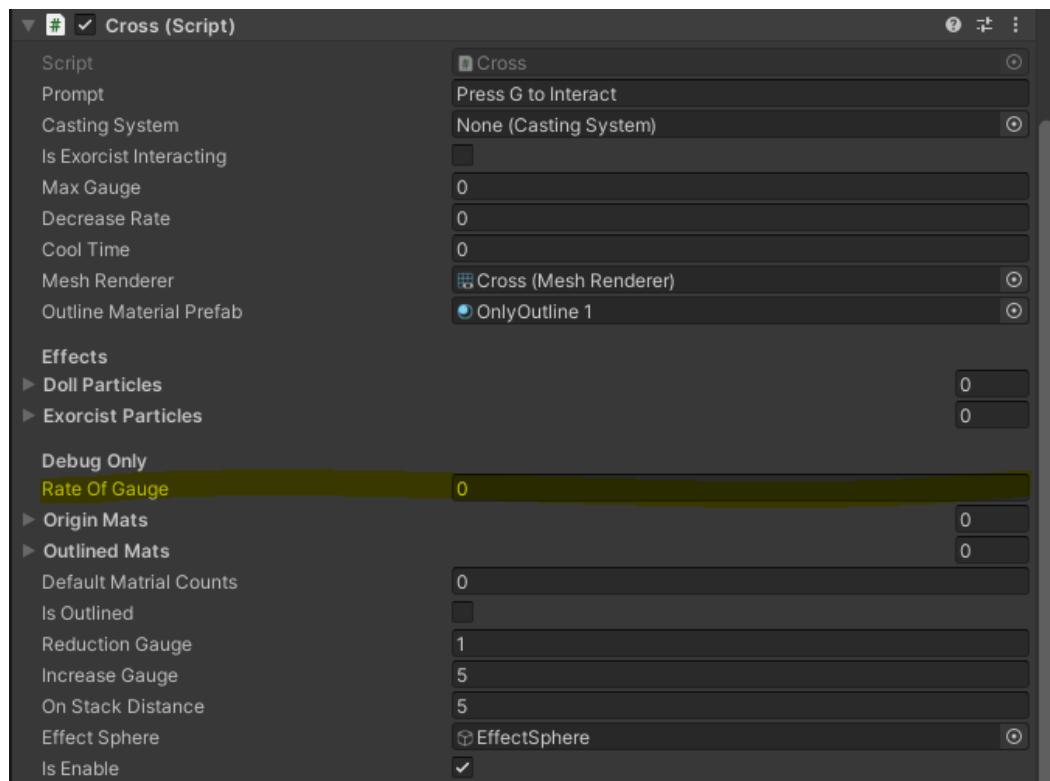
만약 다른 십자가가 존재 시 설치를 할 수 없다는 텍스트를 출력합니다.

화면 중앙에서 레이를 써서 십자가가 감지된다면, 해당 십자가를 회수할 수 있다는 텍스트가 출력됩니다.

회수 텍스트 출력상태에서 스킬 버튼을 누르면 보고 있는 십자가를 회수합니다.



십자가 회수 모습



십자가 게이지

십자가는 게이지를 가지고 있고, 게이지가 0 이 되면 비활성화 상태가 됩니다.

활성화된 십자가에 일정 거리 안으로 인형이 들어오면 인형에게 스택이 쌓입니다.

인형은 이 거리보다 더 먼 곳에서 상호작용을 할 수 있으며 상호작용 시 십자가의 게이지를 줄입니다.



십자가와 상호작용하는 인형의 모습

스택에 따른 인형의 디버프 목록

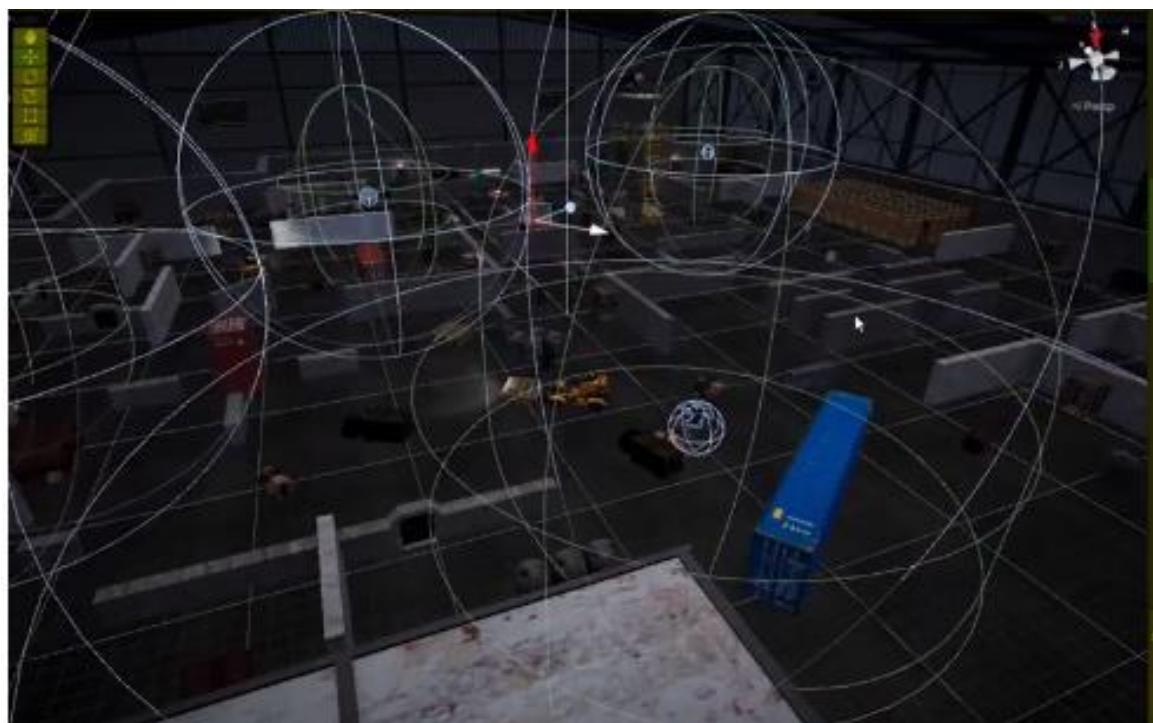
- 1 스택 - 인형이 달릴 때 남기는 흔적의 유지시간이 길어집니다.
- 2 스택 - 인형이 퇴마사에게 받는 데미지가 증가합니다.
- 3 스택 - 퇴마사가 인형을 추격할 때 퇴마사의 이동속도가 증가합니다.
- 4 스택 - 퇴마사 근처에 인형이 있을 경우, 인형의 위치가 노출됩니다.
- 5 스택 - 퇴마사가 인형을 추격할 때 퇴마사의 이동속도가 더 증가합니다.

악령사냥꾼 캐릭터 패시브 스킬 - 까마귀 소환

까마귀는 헌터로 시작하면 게임 시작부터 일정 시간마다 까마귀를 재소환, 및 재배치 그리고 소환되는 개수가 늘어납니다.

까마귀가 소환되는 위치는 랜덤으로 소환될 위치가 정해집니다.

단 랜덤으로 소환되기 때문에, 다른 설치들과 달리, 퇴마사 클라이언트에서 랜덤으로 생성 후, 생성 위치를 다른 클라이언트에게 알려주어야 합니다.



소환된 까마귀의 모습

까마귀 근처에 인형이 일정 시간 동안 있으면 인형은 까마귀 게이지가 올라갑니다.

까마귀 게이지가 일정 수치를 넘어가면 퇴마사에게 감지되며 벽을 넘어 보이게 됩니다.



까마귀의 효과로 인형의 위치를 확인하는 모습



악령사냥꾼 캐릭터 액티브 스킬 - 트랩 설치

헌터는 스킬 버튼을 눌러 트랩을 설치할 수 있습니다.

캐스팅 시스템을 사용하기 때문에 설치 중간에 취소가 가능합니다.



악령사냥꾼이 설치한 뒷의 모습

트랩 위에 Trigger 가 임의의 인형을 감지하면 인형은 상태가 변하고 움직일 수 없게 됩니다.



뒷에 걸린 인형의 모습

인형은 트랩에 임의의 인형이 걸려야 상호작용이 가능하고 트랩의 게이지는 0 이 됩니다.

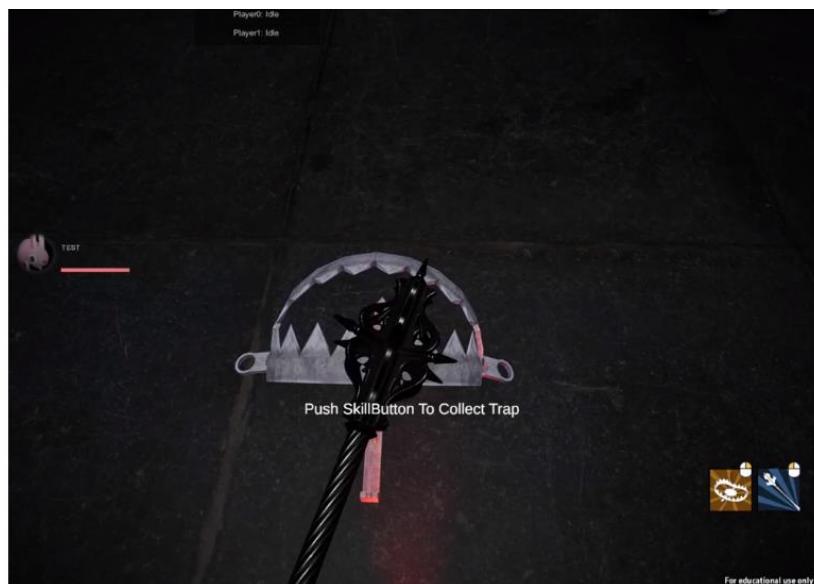
트랩에 걸린 인형은 아래 트랩을 감지하고 상호작용을 통해 게이지를 1로 만들면 트랩에서 빠져나올 수 있습니다.

이 상호작용은 동료 인형도 트랩을 감지하면 가능합니다

트랩은 인형을 한번 잡았었고, 현재 인형이 없으면 회수할 수 있는 상태가 됩니다.

퇴마사는 카메라 중앙에서 레이를 쏘서 회수가 가능한 트랩이 있다면, 회수 가능하다는 텍스트를 출력합니다.

텍스트 출력 시 스킬 버튼을 누르면 해당 트랩을 회수합니다.



사용된 덫을 회수하는 악령사냥꾼의 모습

인형 플레이어 캐릭터



인형 플레이어는 게임 씬에서 다수가 가지는 직업입니다.

인형은 퇴마사의 공격을 피하며, 일반제단들을 활성화, 최종제단까지 활성화 후 탈출하는 것이 목표입니다.

플레이어가 탈출하는 경우 해당 플레이어만 게임에서 승리하게 됩니다.

플레이어가 유령이 되면 해당 플레이어는 패배로 간주하며, 유령으로 관전이 가능합니다.

인형 플레이어는 3 인칭 시점에서 게임 플레이가 진행됩니다.

인형 캐릭터 공통 스킬

숨기

인형은 BvHide 의 상태로 전환됩니다.

주변 가짜 인형처럼 캐릭터 모델의 위치를 변경합니다.



달리기

왼쪽 Shift 을 누르고 이동하면 이동속도가 증가합니다.

달릴 경우 효과들을 스팬하는 스포너가 활성화 됩니다.

활성화된 스포너는 지나간 길에 효과를 만들어 냅니다.

해당 효과는 퇴마사 클라이언트에만 보입니다.



인형이 달릴 때 남기는 흔적

토끼 캐릭터 패시브 스킬 - 이동속도 증가

숨기 상태에서 퇴마사가 일정 거리 안에 있을 경우 활성화되며,

숨기를 풀었을 경우 이동속도 증가 버프를 받습니다.

단, 숨은 상태에서 퇴마사로부터 공격을 받으면 해당 버프는 받지 못합니다.

토끼 캐릭터 액티브 스킬 - 회복

모든 인형은 HealTrigger라는 오브젝트를 보유하고 있습니다.



토끼는 이 HealTrigger를 감지할 수 있습니다.

감지한 HealTrigger를 가진 인형의 인형 HP가 깎인 상태라면 스킬을 사용할 수 있다는 텍스트를 출력합니다.

힐 스킬을 사용하는 토끼 외 피격 인형은 둘 다 움직이지 못합니다.

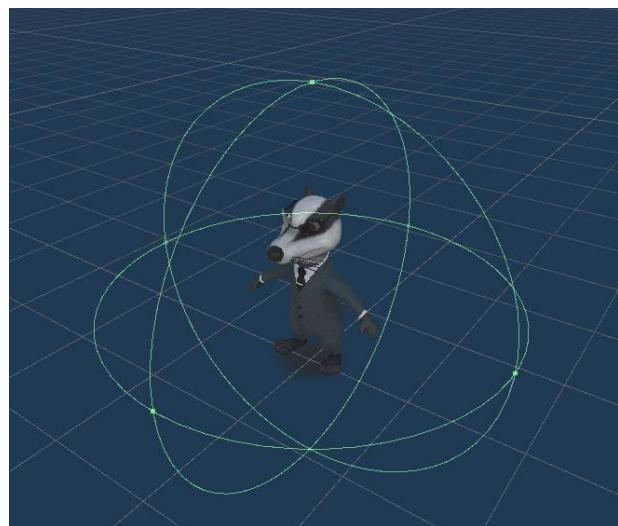
피격 인형은 인형 hp가 회복됩니다.



동료를 회복하는 토끼 인형

늑대 캐릭터 패시브 스킬 - 상호작용 속도 증가

해당 트리거 안에 있는 인형들은 상호작용 속도가 증가합니다.



늑대 패시브 스킬의 작동 범위

늑대 캐릭터 액티브 스킬 - 울부짖기

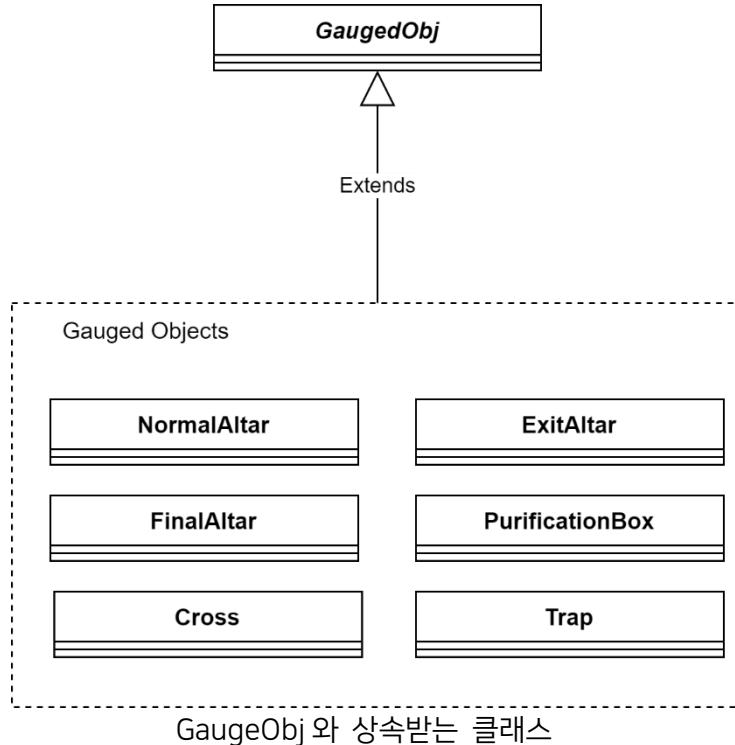
스킬 버튼을 누르면 구 모양의 이펙트가 생기며 주변 퇴마사를 감지하고 드러냅니다.



늑대 액티브 스킬의 작동 모습

오브젝트

공통



Guage 를 가진 오브젝트들은 모두 상속받아야 하는 추상 클래스입니다.

상호작용을 하기 위한 조건 만족 시 화면에 띄울 UI, Effect 를 가지고 있습니다.

가상함수를 상속받음으로써 상호작용을 하기 위한 조건을 오브젝트마다 다르게 설정해줍니다.

CastingSystem 을 가지고 있습니다.

일반 제단

게임 시작 시 정해진 위치 중 위치결정 알고리즘을 통해 생성됩니다.

생성됐을 때 비활성화 상태로 시작하고 게이지는 0입니다.

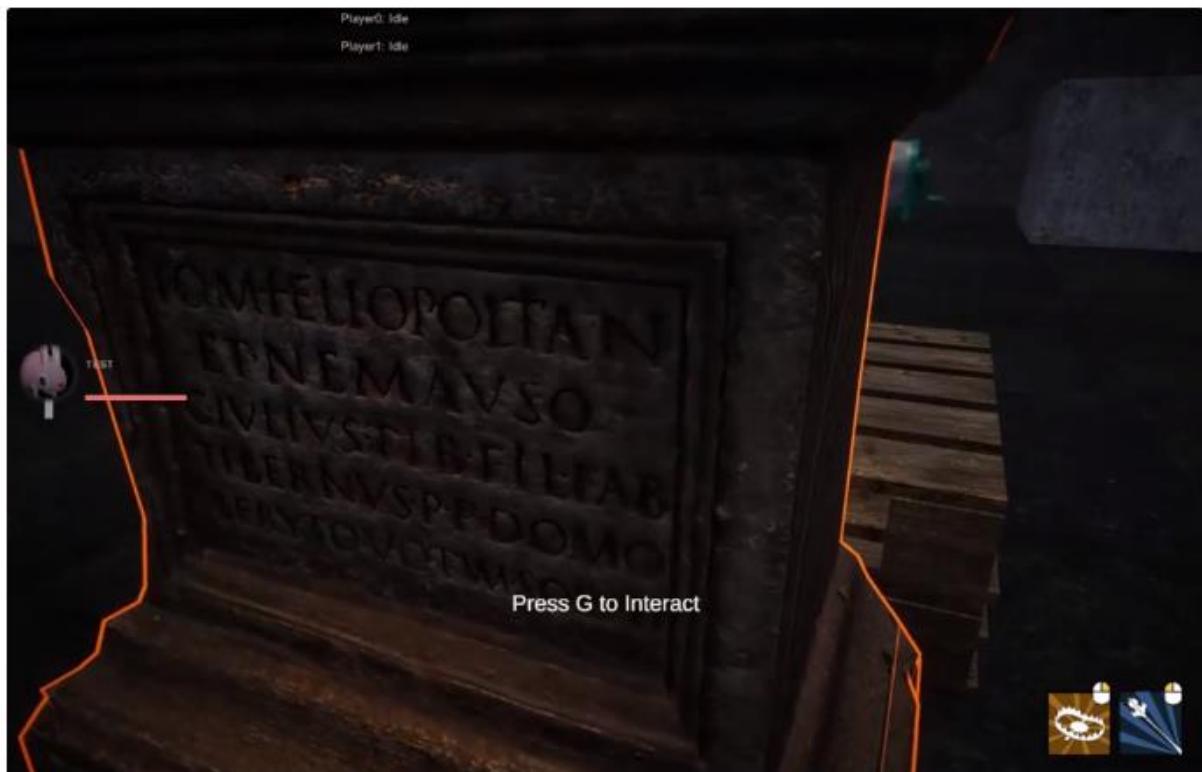
비활성화 상태이며, 게이지가 1 이 아닐 때 인형이 가까이 갈 경우 텍스트와 해당 오브젝트 테두리에 효과가 생깁니다.



일반 제단 상호작용 대기 모습

비활성화 상태의, 게이지가 0.5 이상일 때 퇴마사가 가까이 갈 경우 텍스트와 해당 오브젝트 테두리에 효과가 생깁니다.

이와 같은 효과는 모든 GaugedObj에 적용되었습니다.



상호작용 가능할 때 출력되는 테두리

인형들이 수동 캐스팅을 통해 게이지를 1로 만들면 활성화 상태로 됩니다.

활성화된 NormalAltar는 인형, 퇴마사 둘 다 상호작용이 불가능합니다.



인형이 일반 제단과 상호작용하는 모습

비상 탈출 제단

게임 시작 시 정해진 위치 중 랜덤으로 정해집니다.

퇴마사와 인형 모두의 인원이 2 명 초과라면 나타나지 않습니다.

퇴마사와 인형 모두 인원이 2 명 이하라면 나타나고, 상호작용이 가능해집니다.

인형은 오토 캐스팅을 시작하고 캐스팅이 끝나면 탈출이 가능합니다.



인형이 비상 탈출 제단과 상호작용하는 모습

퇴마사는 수동 캐스팅을 끝내면 오브젝트를 사라지게 합니다.

이후 다른 플레이어들은 상호작용이 불가능합니다.



퇴마사가 탈출 제단과 상호작용하는 모습

최종 탈출 문

게임 시작 시 정해진 위치에 생성됩니다.

문의 역할을 수행하고, 게임 생성시 닫혀있습니다.



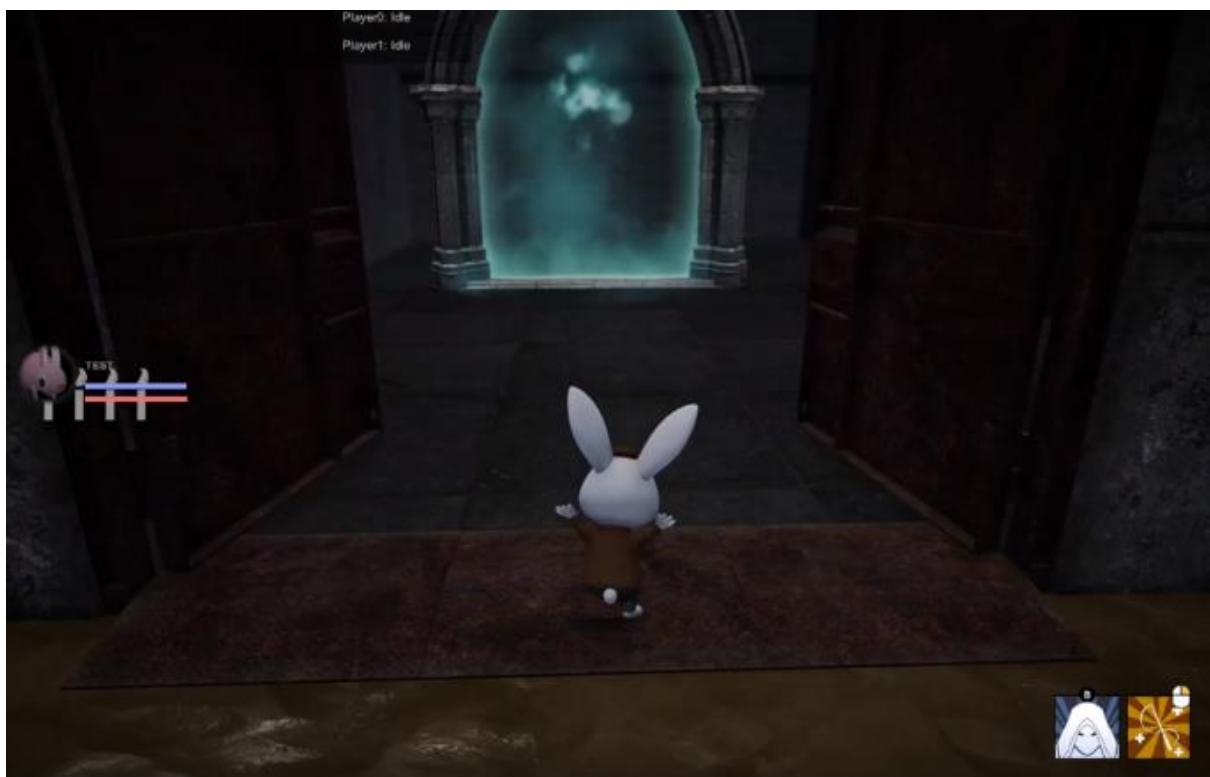
일반 제단 4 개가 모두 활성화되지 않은 경우, 상호작용 불가능한 모습



일반 제단이 모두 활성화된 경우 인형과 상호작용 가능한 모습

인형은 수동 캐스팅을 통해 게이지를 1로 만들 수 있습니다.

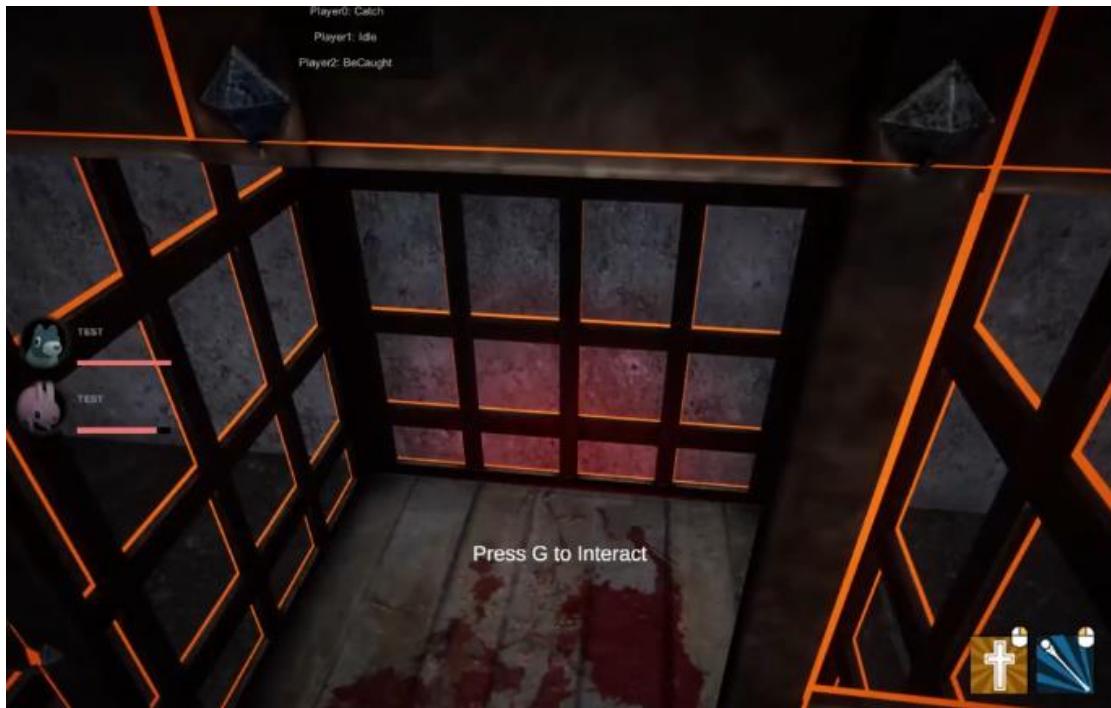
게이지가 1이 된다면 문이 열립니다.



정화 상자

게임 시작 시 정해진 위치에 모두 생성됩니다.

퇴마사가 BvCatch 상태 일 때 퇴마사와 상호작용이 가능합니다.



퇴마사가 인형을 든 상태에서 상호작용 가능한 모습

퇴마사가 오토 캐스팅을 통해 BvBeCaught 인형을 가둘 수 있습니다.

인형을 가두면 게이지는 0 이 됩니다.

가둬진 인형은 BvBePurifying 이 된 후, 매 초 악령 HP 가 줄어듭니다.

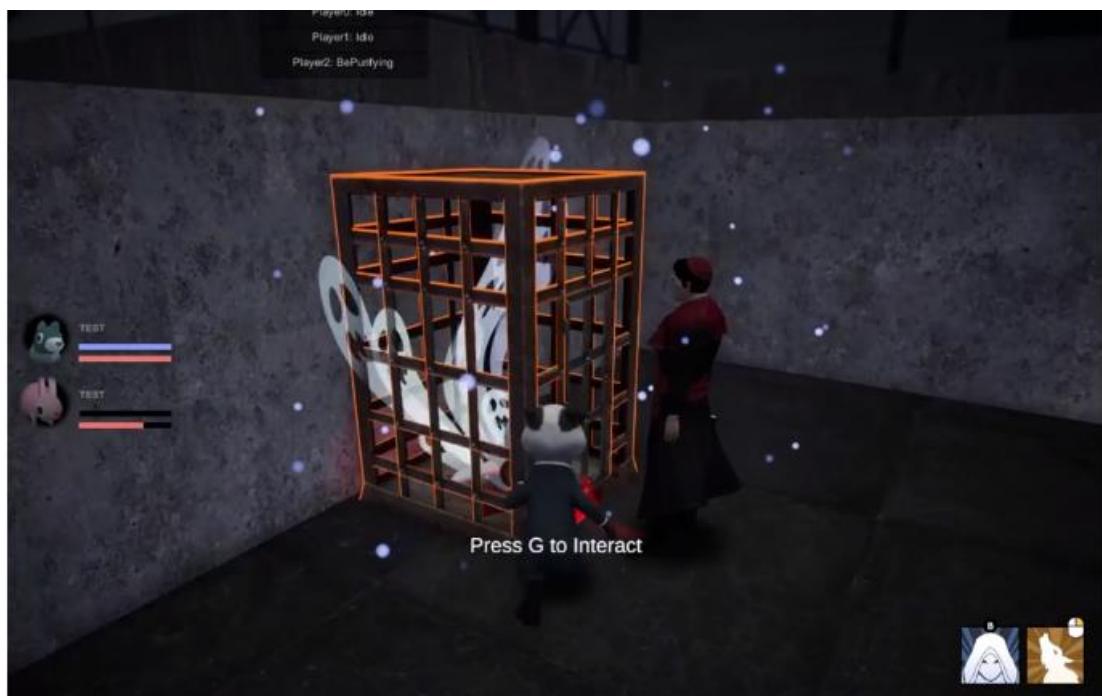


정화 상자에서 정화되는 인형의 모습

인형이 가두어졌을 경우 BvIdle 인 인형과 상호작용이 가능해집니다.

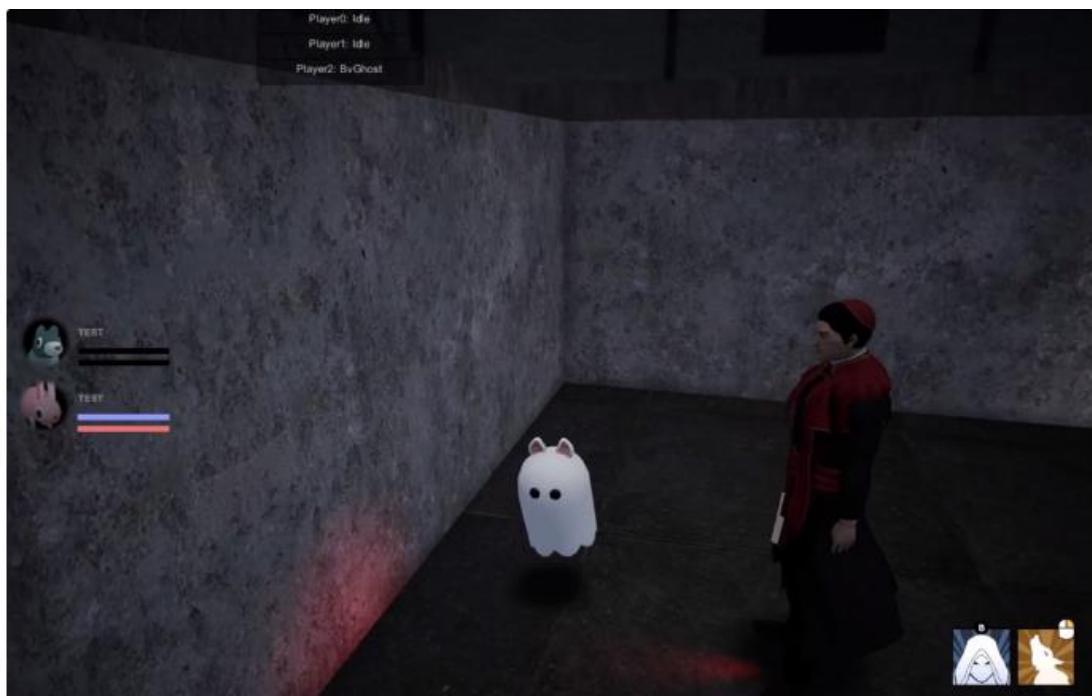
BvIdle 인 인형은 수동 캐스팅을 통해 게이지를 1로 만들 수 있습니다.

게이지가 1이 된다면 갇힌 인형은 PurificationBox에서 벗어나고, BvEscape 상태가 됩니다.



정화 중인 정화 상자와 상호작용 가능한 모습

악령 HP 가 0 이 될 때까지 벗어나지 못한다면, 해당인형은 BvGhost 가 되어 자신을 제외한 다른 플레이어들에게 보이지 않게 됩니다.



인형이 정화되어 유령이 된 모습

자동문

인형만 상호작용이 가능합니다.

게이지가 존재하지 않지만 GaugedObj를 상속하고 있습니다.

오토 캐스팅을 통해 문을 열거나 닫을 수 있습니다.

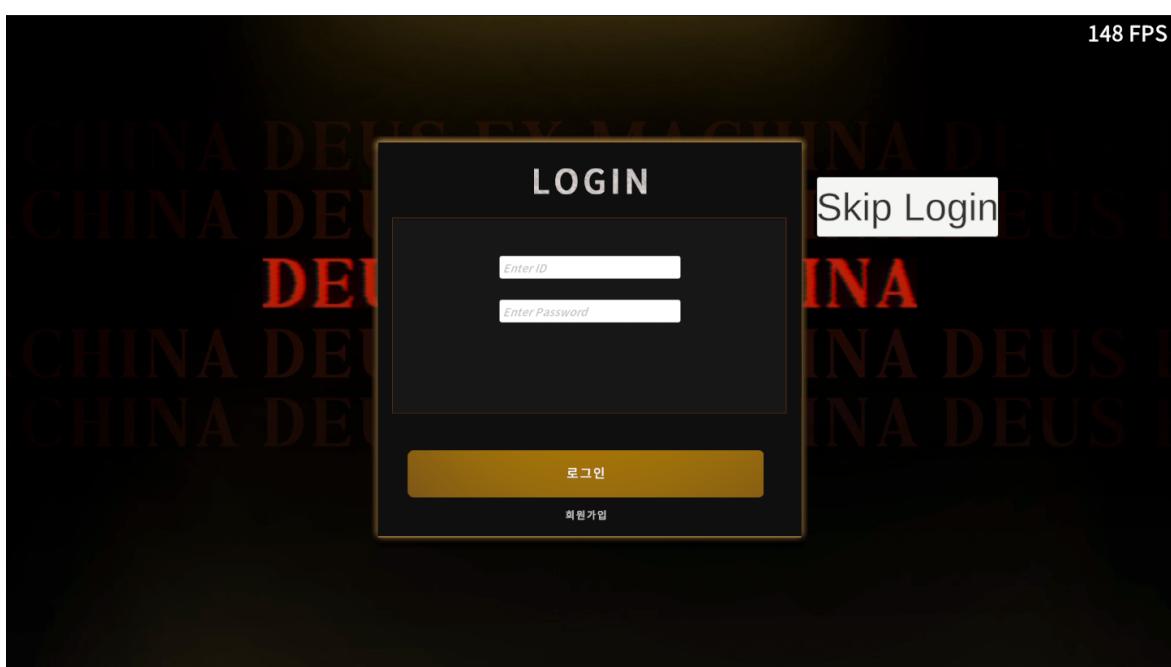
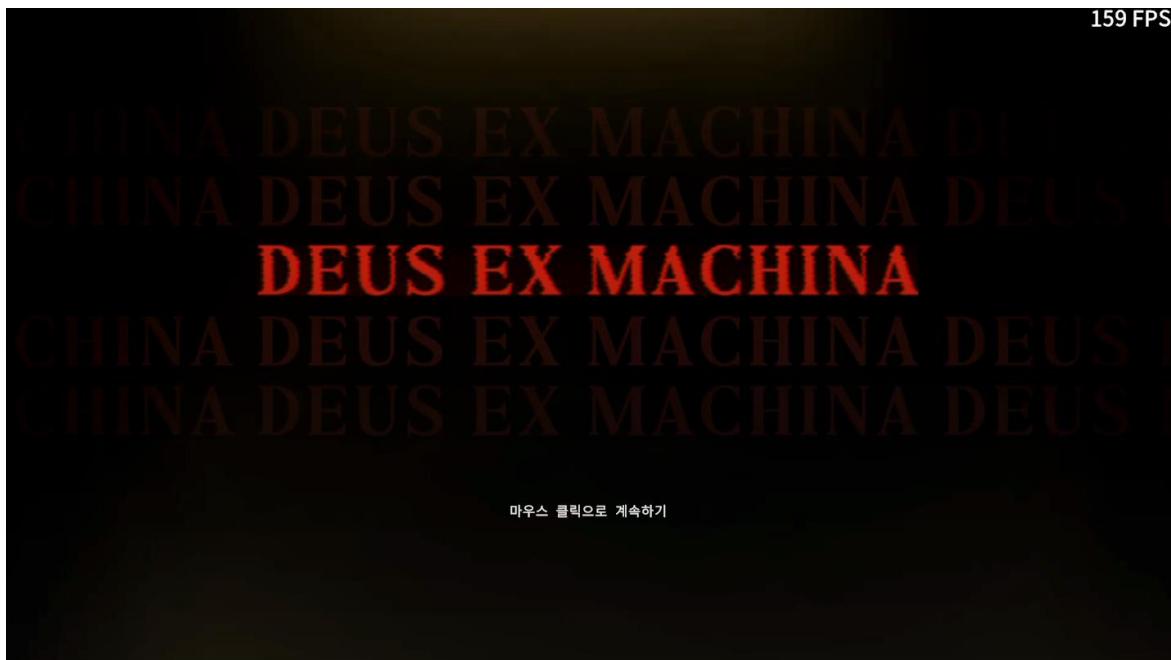
문의 크기 때문에 인형만 드나들 수 있습니다.



자동문과 상호작용 가능한 모습

Scene 소개

00 Launcher



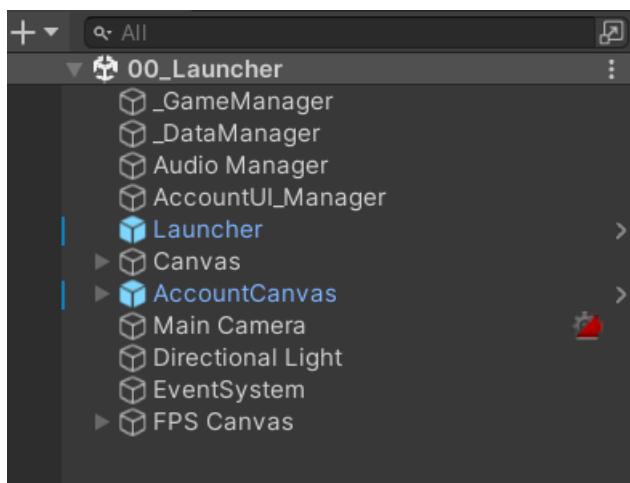
역할

Launcher Scene 은 Photon Network 2 서버와 연결하는 역할을 수행합니다.

서버와 연결된 후 알림 메시지("마우스 클릭으로 계속하기")를 페이드 인-아웃으로 출력합니다.

마우스를 클릭하면 로그인을 할 수 있는 로그인 창이 뜹니다.

Hierarchy



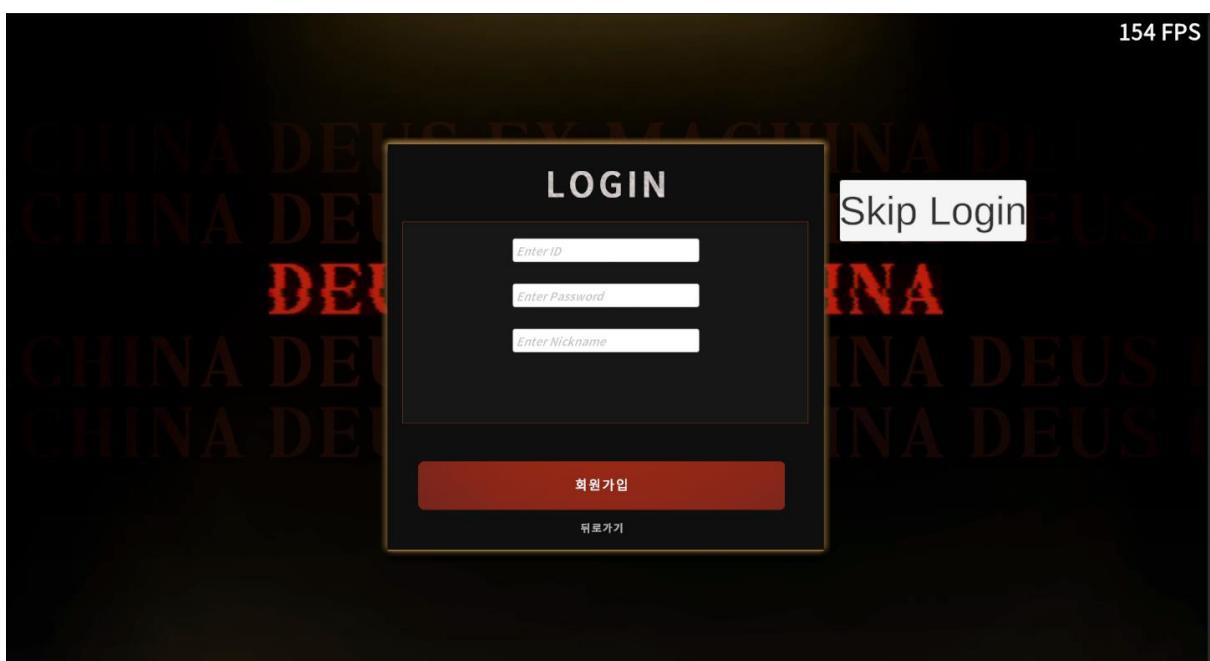
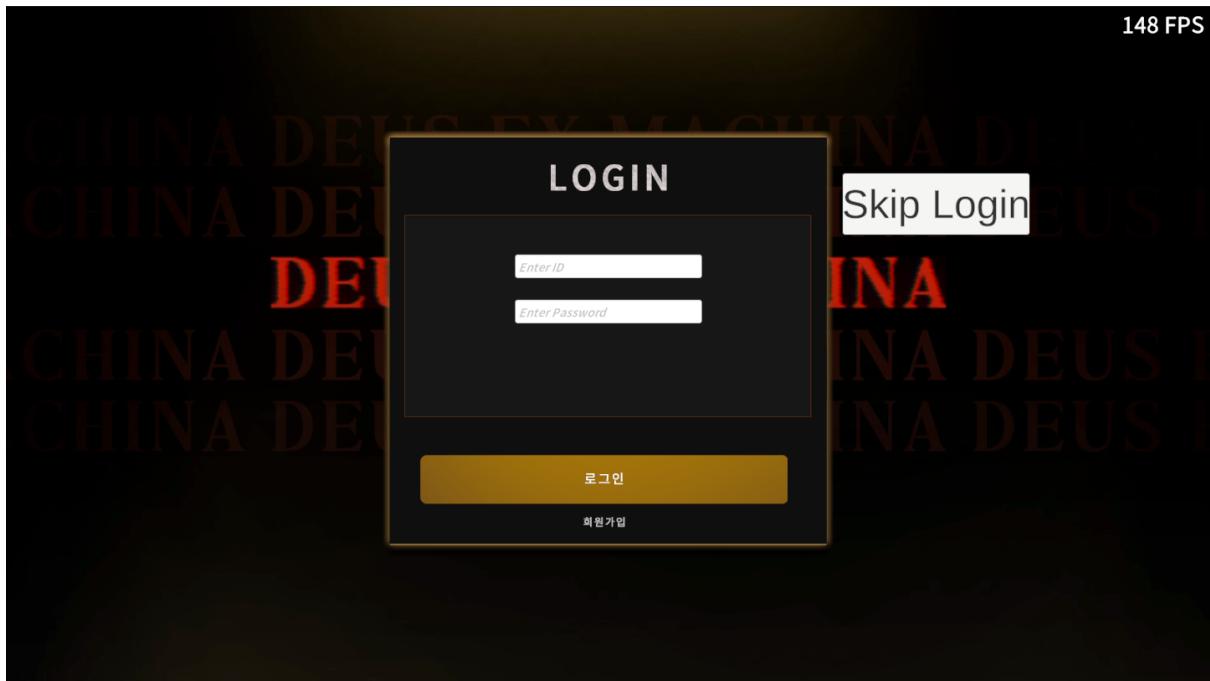
Launcher: 서버 연결을 위한 오브젝트

FPS Canvas: 프레임 출력을 위한 캔버스

_DataManager: 사용자 정보를 저장한다.

AccountCanvas: 사용자 계정 처리 관련 UI Canvas

00 Launcher_LOGIN



역할

사용자 로그인 및 회원가입을 위한 장면입니다.

로그인 및 회원가입 버튼을 누르면 DB 와 연동하여, 해당 작업을 수행합니다.

Skip Login 은 빠른 디버그를 위한 임시 버튼입니다.

02_MainLobbyScene



역할

사용자가 로고 화면 이후 플레이어가 보게 되는 화면입니다.

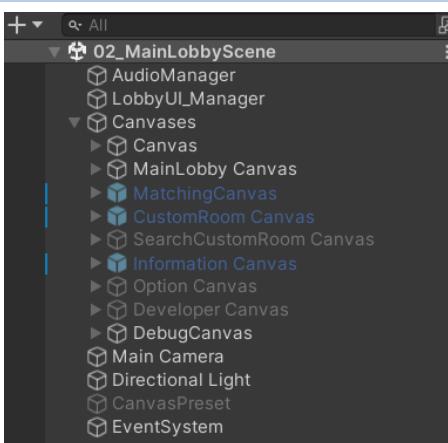
사용자는 게임 플레이를 선택하거나 정보를 확인할 수 있습니다.

사용자는 옵션을 통해 게임의 설정을 변경할 수 있습니다.

사용자는 나가기 버튼을 이용해서 게임을 종료할 수 있습니다.

모든 버튼은 마우스 클릭으로 사용합니다.

Hierarchy



Quick Match Buttons Panel



역할

사용자가 빠른 매칭을 눌렀을 때 보여지는 화면으로 사용자가 역할을 선택해 매칭을 할 수 있도록 만든 유저 인터페이스입니다.

인형으로 플레이: 사용자가 인형의 역할로 게임을 플레이하고 싶을 경우 선택하는 버튼입니다.

퇴마사로 플레이: 사용자가 퇴마사의 역할로 게임을 플레이하고 싶을 경우 선택하는 버튼입니다.

돌아가기: 사용자가 플레이하고 싶지 않을 경우 혹은 잘 못 선택할 경우 뒤로 돌아가는 화면입니다. 돌아가기 버튼을 누르면 바로 전 화면으로 돌아갑니다.

Matching Canvas



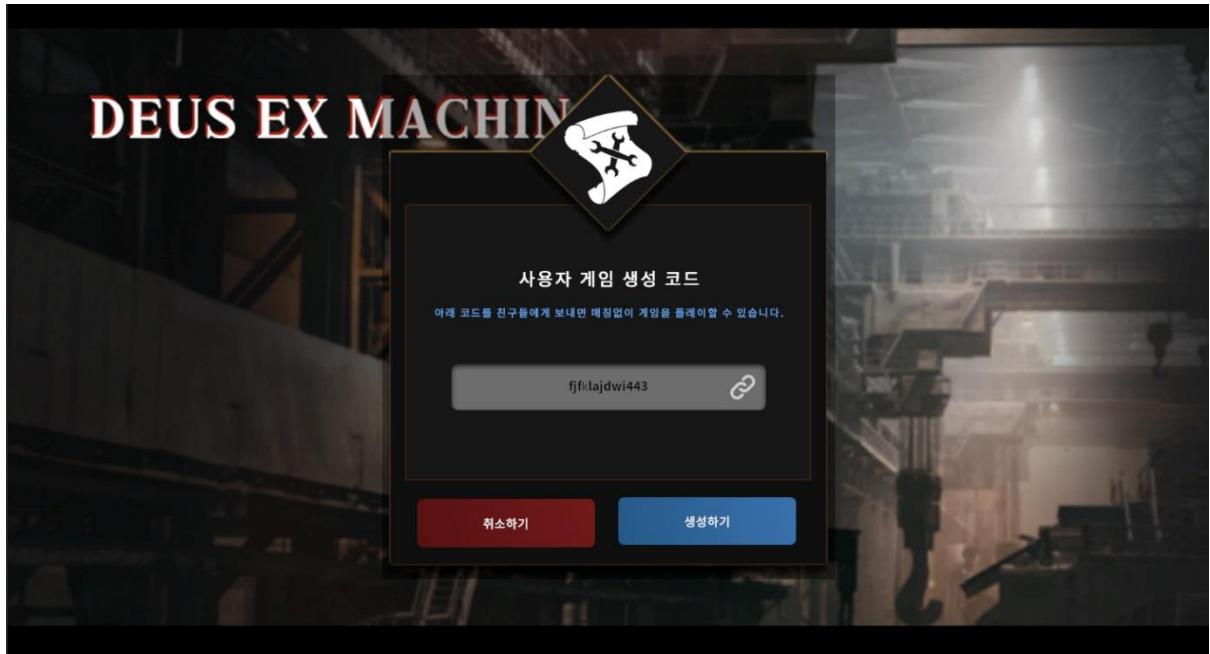
역할

빠른 매칭으로 들어온 플레이어들이 보는 화면입니다.

매칭 취소: 매칭이 취소되는 버튼으로 매칭을 원하지 않을 경우 매칭 취소 버튼을 누르면 취소 한다. 취소 버튼을 누를 경우 바로 전 화면으로 돌아갑니다.

Skip Button: 개발자 모드 버튼입니다. 개발을 편리하게 하기 위해 만든 임시 버튼입니다.

CustomRoom Canvas



역할

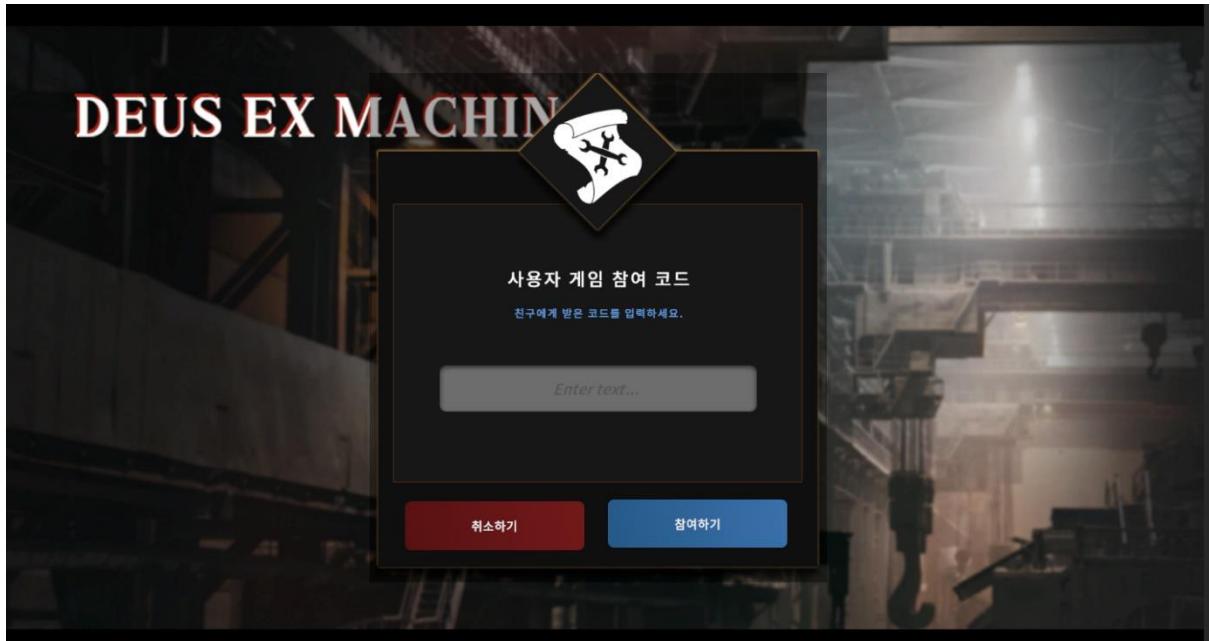
사용자가 게임을 생성하는 캔버스입니다. 게임 생성 코드가 랜덤으로 생성되고 복사 버튼을 눌러서 복사할 수 있습니다. 사용자 게임을 생성하고 싶지 않으면 취소하기를 눌러서 취소할 수 있습니다. 생성하기를 누르면 사용자 생성 게임이 생성됩니다.

 : 사용자 게임 생성 코드를 복사하는 버튼. 버튼을 누르면 클립보드에 생성 코드를 복사합니다.

생성하기: 게임 생성 코드를 이용해 사용자 게임을 만듭니다.

취소하기: 게임 생성을 취소하고 메인 로비로 되돌아가는 버튼입니다.

Search Custom Panel



역할

사용자가 이미 생성된 게임에 참여하기 위한 화면입니다

화면 설명

Enter text...: 전달 받은 코드를 입력하는 곳입니다.

참여하기: 알맞은 코드를 입력하고 참여하기를 누르면 사용자가 생성한 방에 참여할 수 있습니다.

취소하기: 게임에 참여하지 원하지 않거나 잘못된 버튼을 눌렀을 경우 되돌아 갈 수 있는 버튼입니다.

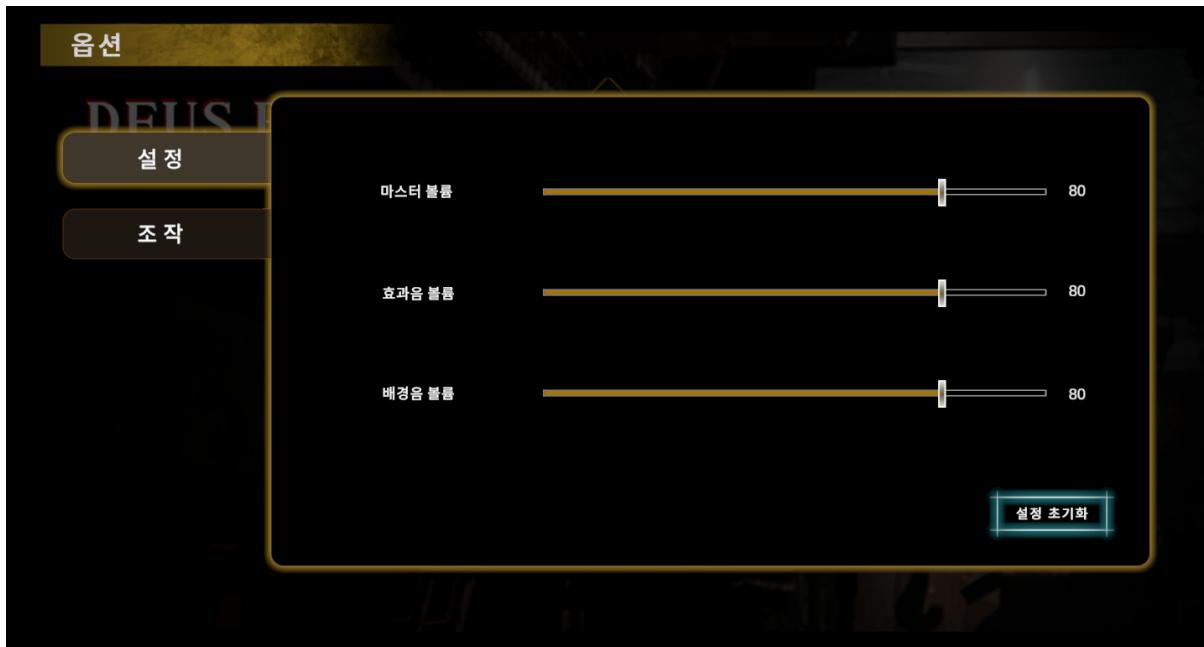
Information Panel



역할

게임에 관한 정보를 선택하여 볼 수 있습니다. 확인할 수 있는 정보는 인형, 퇴마사 정보가 있습니다.

Option Panel



역할

게임 옵션을 조절할 수 있는 화면입니다. 설정에서는 게임의 볼륨을 조절할 수 있습니다.

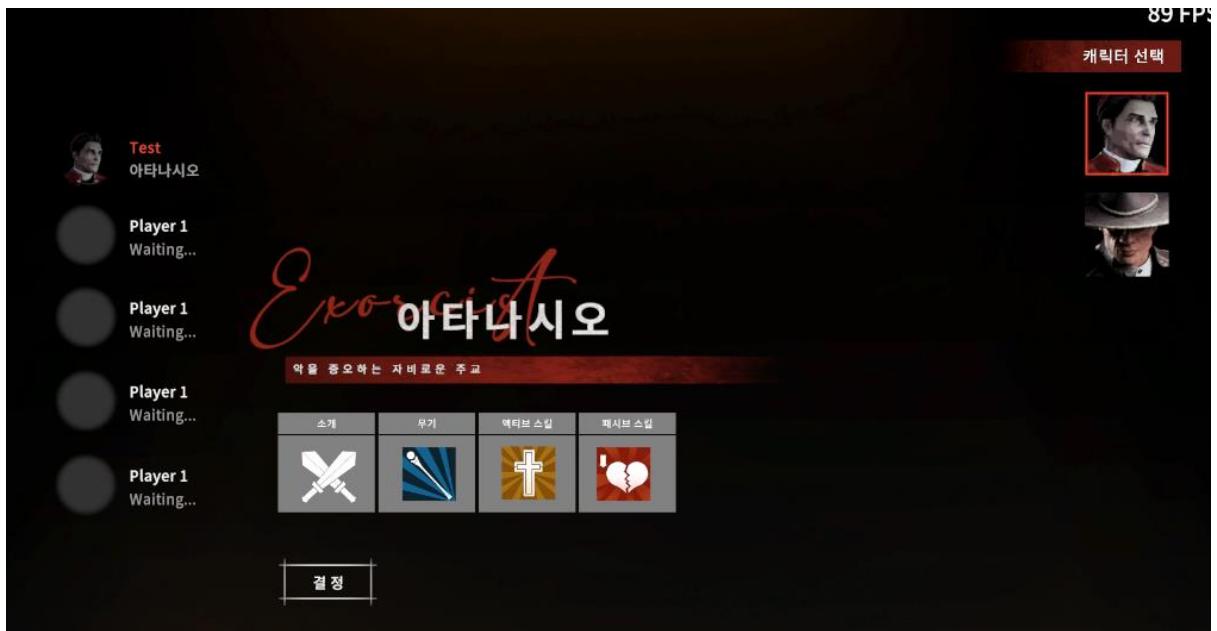
화면 설명

마스터 볼륨: 전체 음량을 줄이거나 늘릴 수 있습니다. 조절할 수 있는 값은 0~100입니다.

효과음 볼륨: 게임의 효과음을 줄이거나 늘릴 수 있습니다. 조절할 수 있는 값은 0~100입니다.

배경음 볼륨: 게임의 배경음을 줄이거나 늘릴 수 있습니다. 조절할 수 있는 값은 0~100입니다.

03_SelectCharacterScene



역할

역할을 선택한 매칭 화면 이후에 캐릭터를 선택하는 화면입니다.

역할에 따라 서로 다른 화면을 보게 됩니다.

오른쪽 세로로 나열 되어있는 버튼을 이용해서 캐릭터를 선택할 수 있습니다.

화면의 가운데에는 캐릭터에 대한 설명이 있습니다.

왼쪽 세로로 나열 되어있는 UI를 다른 사람들이 무슨 캐릭터를 선택하는지 확인할 수 있습니다.

플레이어 본인은 항상 상단에 위치하며 플레이어가 인형일 경우, 퇴마사는 항상 최하단에 위치합니다. 퇴마사일 경우에는 플레이어의 순서대로 나열됩니다.

03_RoomScene



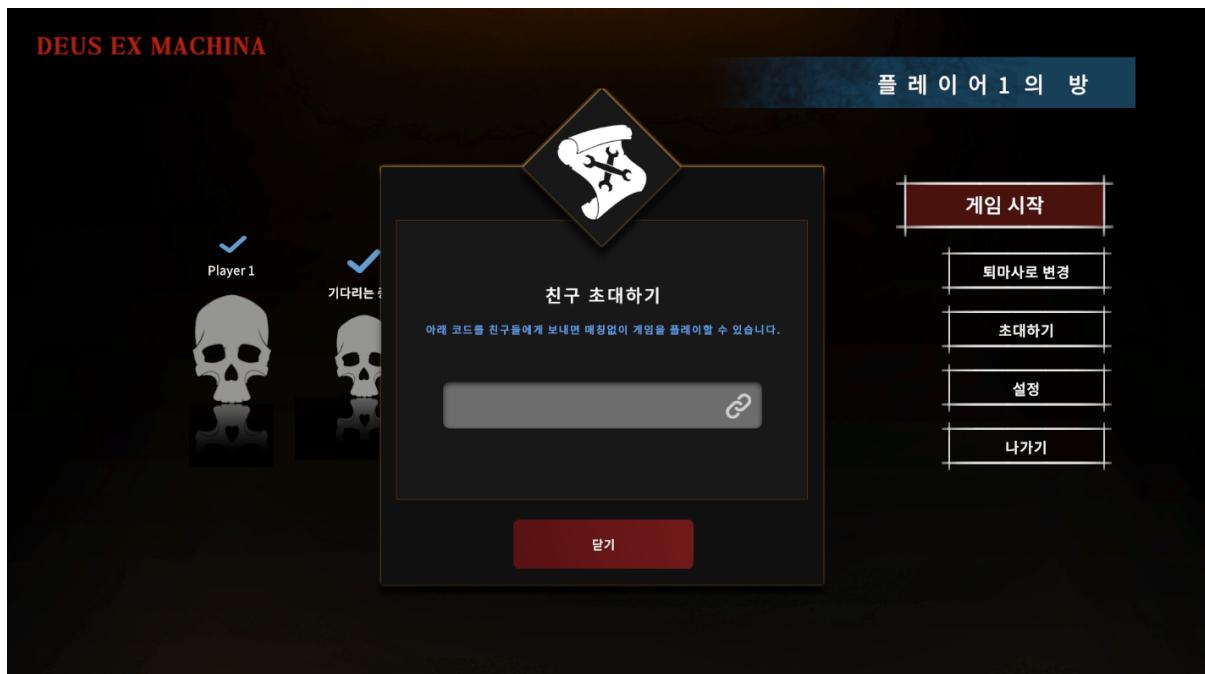
역할

사용자가 커스텀 룸을 만들 경우 보게 되는 화면입니다.

게임 시작: 게임을 시작할 수 있는 버튼. 커스텀 룸을 생성한 플레이어만이 게임을 시작할 수 있습니다.

퇴마사로 변경: 역할을 변경할 수 있는 버튼으로 퇴마사일 경우 버튼을 누르면 인형으로 변경되며, 반대의 경우 인형에서 퇴마사로 역할이 변경됩니다.

초대하기 | Canvas



화면 설명

초대하기: 다른 사람을 커스텀 룸에 초대할 때 사용하며 커스텀 룸을 생성할 때 사용한 코드를 복사해 다른 사람을 초대할 수 있습니다.

설정: 게임의 옵션을 설정하는 버튼입니다.

나가기: 커스텀 룸에서 나가는 버튼입니다. 나가기 버튼을 누를 경우 메인 로비로 되돌아갑니다.

MainGameScene



퇴마사 화면(1 인칭)

퇴마사 UI

- 왼쪽 가운데: 인형 체력 바
- 오른쪽 아래: 스킬 UI
- 퇴마사 조작법
- 이동: wsad, 마우스이동
- 광폭화: 마우스 오른쪽버튼
- 공격: 마우스 왼쪽버튼
- 쓰러진 인형줍기: 마우스 왼쪽버튼
- 오브젝트와 상호작용: 일정 조건 만족 시 키보드 G 키 버튼

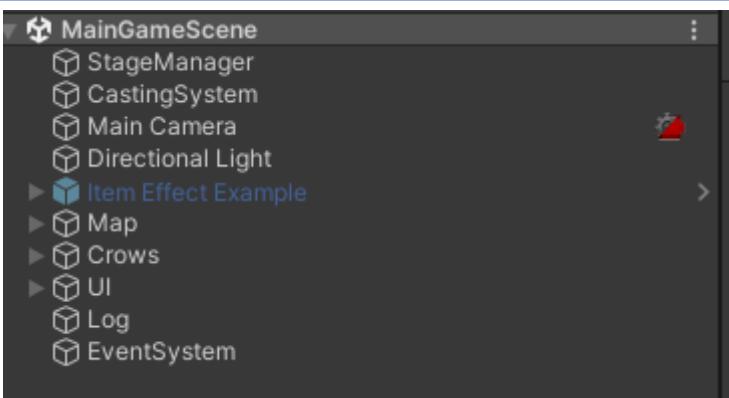


인형 화면(3 인칭)

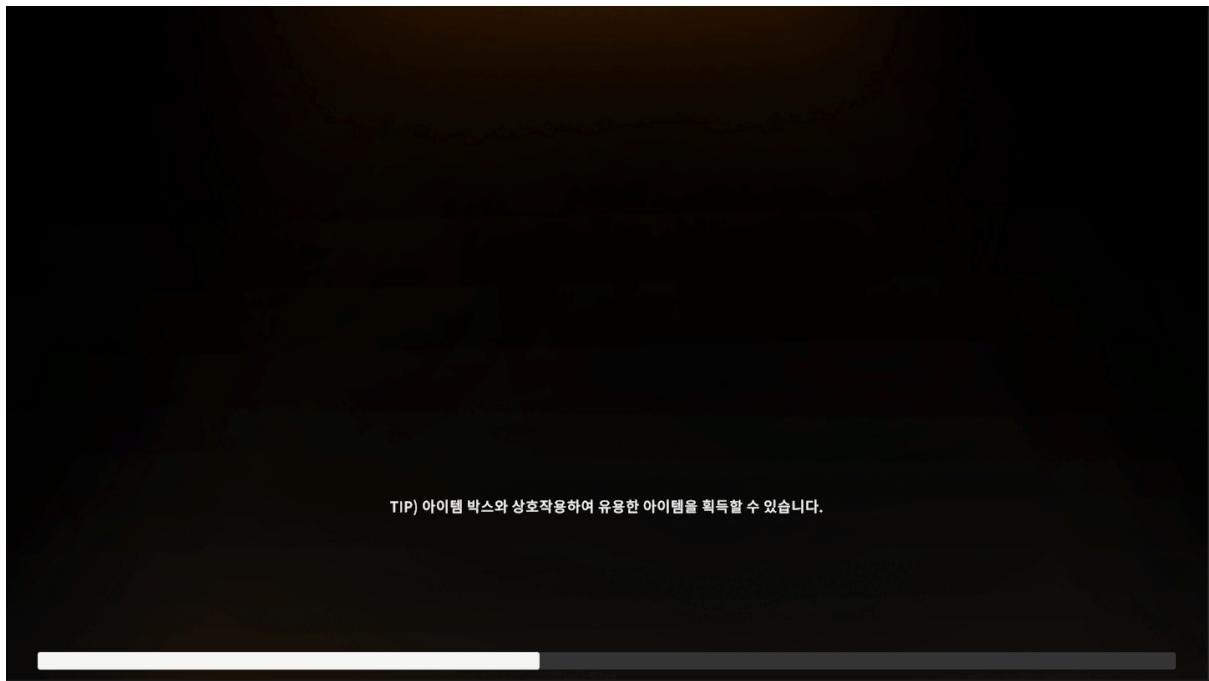
인형 UI

- 왼쪽 가운데: 인형 HP 바
- 오른쪽 아래: 스킬 UI
- 인형 조작법
- 이동: wasd, 마우스 이동
- 대쉬: LShift
- 스킬: 오른쪽 마우스 버튼
- 물체 작동: 키보드 G 키

Hierarchy



97_LoadingScene

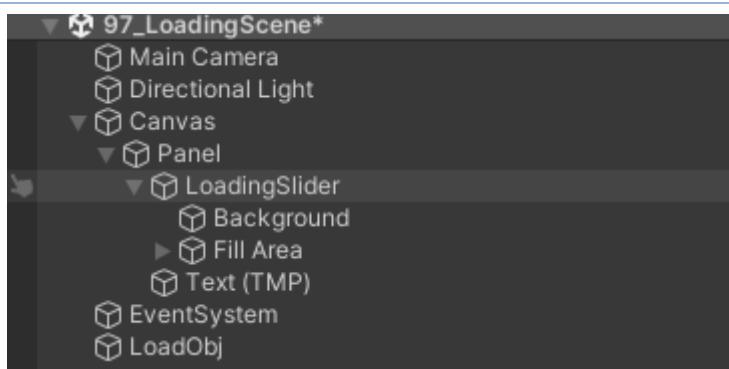


역할

Tip.: 게임 플레이에 관해 설명하는 문구가 나타납니다.

Slider: 로딩의 진행 상황을 알려주는 역할입니다.

Hierarchy



98_LoadingNetworkScene

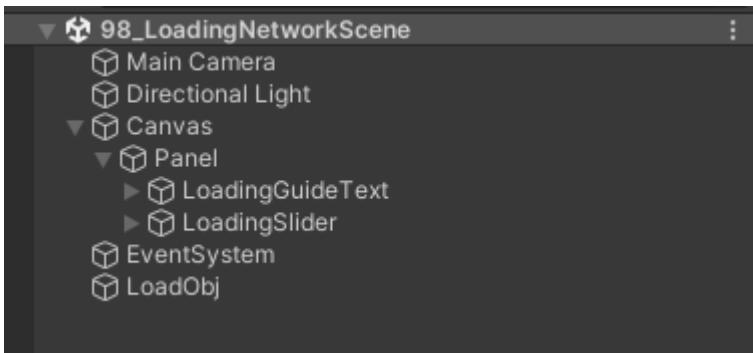


역할

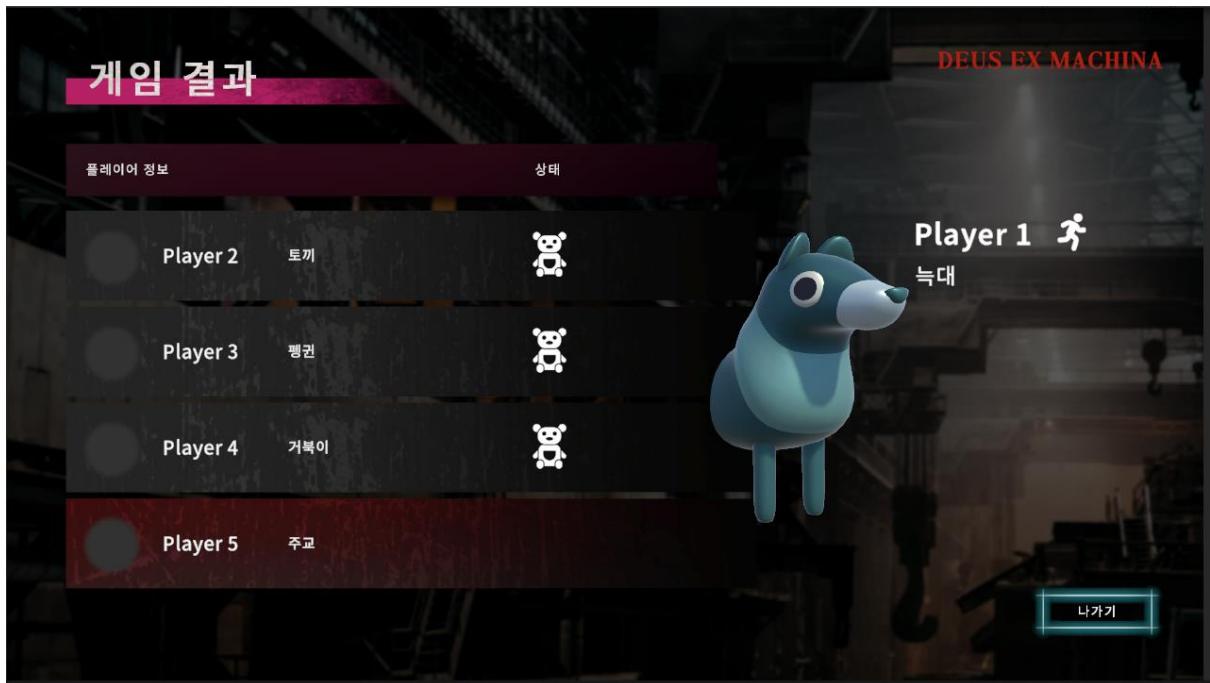
인형 공장: 게임의 배경 정보에 관한 문구가 나타납니다.

Slider: 로딩의 진행 상황을 알려주는 역할입니다.

Hierarchy



99_GameResultScene



역할

게임 결과 화면입니다.

왼쪽은 플레이어 자신의 결과 화면이며, 오른쪽은 다른 플레이어들의 결과를 표시합니다.

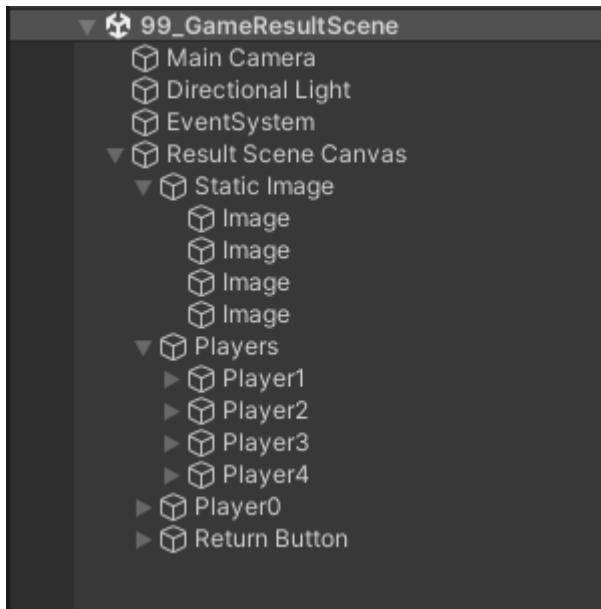
표시되는 정보는 플레이어의 이름, 역할, 상태입니다.

자신의 결과화면에는 이름, 역할, 상태, 그리고 사용한 캐릭터의 3D 모델을 배치합니다.

사용자의 상태는 이미지로 나타냅니다. 위의 화면에서 곰 인형은 Alive 상태를 나타내며 사람 모양의 이미지는 탈출 상태를 나타냅니다.

나가기: 오른쪽 아래의 나가기 버튼을 누를 경우, 메인 로비로 돌아갑니다.

Hierarchy



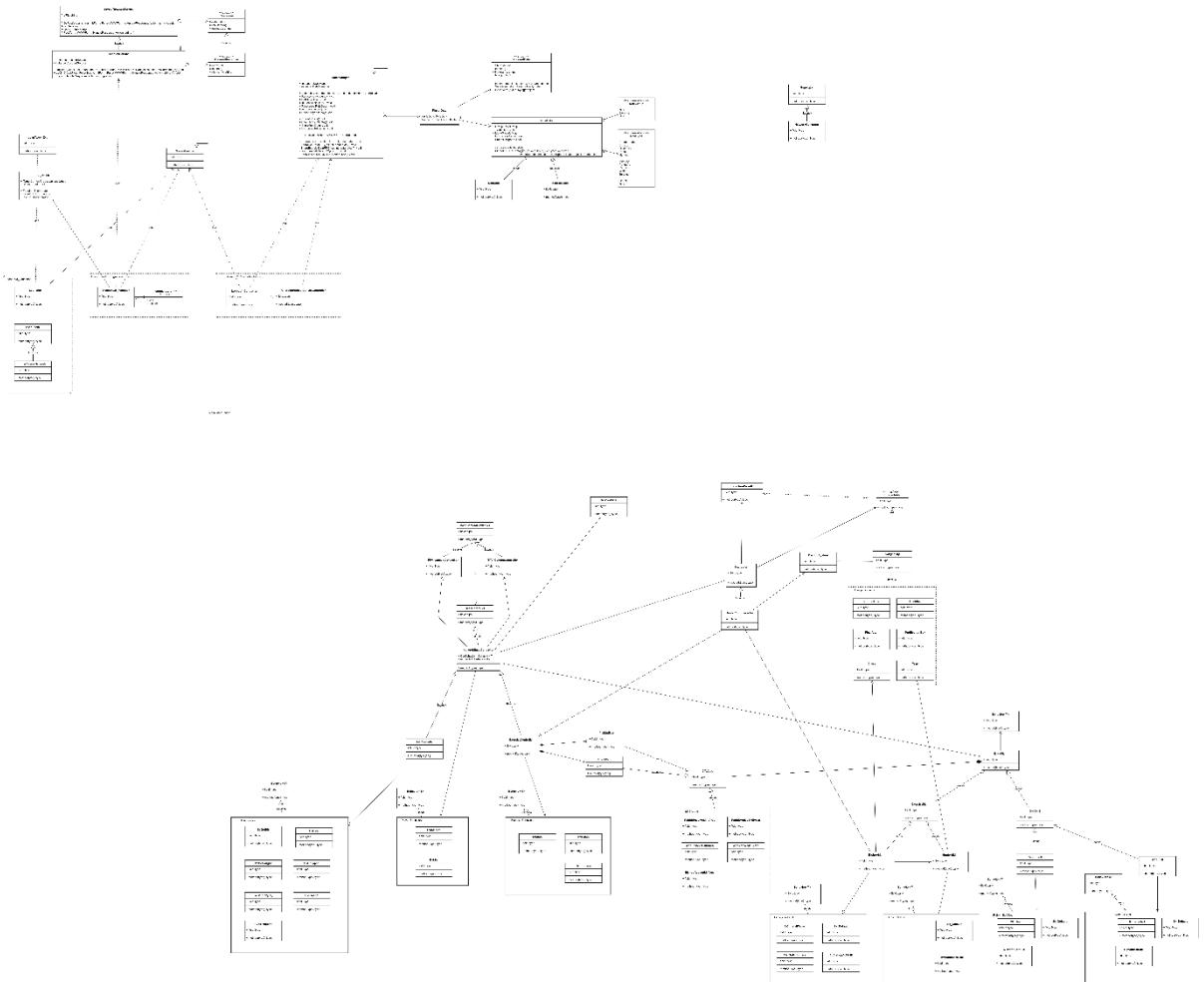
- Static Image: 상호작용이 없는 이미지를 모아 놓은 것입니다. 게임의 로고와 배경화면과 게임 결과 로고 이미지가 있습니다.
- Players: 플레이어의 이름과 역할, 역할 아이콘, 상태 아이콘을 하나로 만들어 놓은 오브젝트입니다.
- Player0: 화면 오른쪽의 Player1의 텍스트와 이미지, 모델을 하나로 만들어 놓은 오브젝트입니다.
- Return Button: 화면에서 나가기 위한 버튼입니다.

Asset List

Asset	Type	Link
Animation_Asset	퇴마사 애니메이션	mixamo
Character_Asset	퇴마사, 주교	https://assetstore.unity.com/packages/3d/characters/humanoids/humans/priest-193848#content
	퇴마사, 신부	https://assetstore.unity.com/packages/3d/characters/humanoids/humans/priest-193848#content
	퇴마사, 심령사진가	https://assetstore.unity.com/packages/3d/characters/humanoids/humans/mia-pbr-character-231569
	퇴마사, 악령사냥꾼	https://assetstore.unity.com/packages/3d/characters/humanoids/humans/3d-dark-warrior-227076

Character Object Asset	퇴마사 주교 무기, 지팡이	https://assetstore.unity.com/packages/3d/props/weapons/fantasy-staff-set-05-194332
	퇴마사 신부 무기, 단검	https://assetstore.unity.com/?q=dagger&orderBy=1
	퇴마사 악령 사냥꾼, 메이스	https://assetstore.unity.com/packages/3d/props/weapons/mace-20-191209
	퇴마사 주교,	https://assetstore.unity.com/packages/3d/props/interior/books-pack-5484
	퇴마사 악령 사냥꾼, 새	https://assetstore.unity.com/packages/3d/characters/animals/birds/living-birds-15649
	퇴마사 악령 사냥꾼,	https://assetstore.unity.com/packages/3d/props/weapons/animated-3d-trap-8288
	퇴마사 주교, 십자가	https://assetstore.unity.com/packages/3d/props/exterior/grave-cross-tombstone-11085
CharacterEffect	캐릭터 이펙트	https://assetstore.unity.com/packages/vfx/particles/magical-orb-133296 Hovl Studio
CustomMap	-	-
Font_Asset	폰트	Noto_Sans
UI_Asset	-	-

Class Diagram



전체 클래스 다이어그램