

# 캡스톤 디자인 I 종합설계 프로젝트

프로젝트 명	Ai Senma
팀 명	LAB 36
문서 제목	중간보고서

Version	1.0
Date	2024-03-29

팀원	문민철(팀장)
	강기범
	강동우
	김현준
	박세호
	이재원
	오정규
지도교수	김 장 호 교수

 <div>국민대학교 소프트웨어학부 캡스톤 디자인 I</div>	중간보고서		
	프로젝트 명	Ai Senma	
	팀 명	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 "Ai Senma"를 수행하는 팀 "LAB 36"의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 "LAB 36"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역

Filename	중간보고서-LAB 36.doc
원안작성자	오정규
수정작업자	

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2024-03-28	오정규	1.0	신규작성	신규작성

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
	프로젝트 명	Ai Senma	
	팀 명	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

## 목 차

1	프로젝트 목표 .....	4
2	수행 내용 및 중간결과 .....	5
2.1	계획서 상의 기획내용 .....	5
2.1.1	팀 운영 방식 .....	5
2.1.2	게임 배경 및 스토리 .....	6
2.1.3	카메라 및 플레이어 이동 .....	8
2.1.4	퍼즐 기믹 구성 - 염력 및 크기 조정 .....	9
2.1.5	퍼즐 기믹 구성 - 중력 .....	10
2.1.6	퍼즐 기믹 구성 - 빛 .....	11
2.1.7	멀티 플레이 .....	12
2.1.8	맵 .....	13
2.2	수행내용 .....	14
2.2.1	프로젝트 진행 내용 - 플레이어 및 카메라 개발 .....	14
2.2.2	프로젝트 진행 내용 - 염력 및 크기 조정 .....	15
2.2.3	프로젝트 진행 내용 - 중력 .....	16
2.2.4	프로젝트 진행 내용 - 빛 .....	18
2.2.5	프로젝트 진행 내용 - 네트워크 동기화 기능 .....	19
3	수정된 개발 내용 및 개발 과정 .....	22
3.1	수정사항 및 문제해결 과정 .....	22
3.1.1	플레이어 및 카메라 개발 .....	22
3.1.2	염력 및 크기 조정 .....	22
3.1.3	중력 .....	23
3.1.4	빛 .....	24
3.1.5	네트워크 개발 .....	24
4	향후 추진계획 .....	25
4.1	향후 계획의 세부 내용 .....	25

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	<b>Ai Senma</b>	
	<b>팀 명</b>	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

## 1 프로젝트 목표

최근 게임업계에서는 배틀그라운드와 오버워치 그리고 리그오브레전드와 같은 멀티플레이 게임들이 흥행하고 있습니다. "더로그(The Log)"라는 게임 통계 서비스의 3월 26일자 사용자 기준 점유율 10위의 게임은 전부 멀티플레이 게임입니다.

또한 소규모 게임회사 혹은 팀에서 개발한 게임일지라도 게임성과 멀티플레이 가능여부 그리고 아이디어에 따라서 과거와 달리 구현의 완성도나 아름다운 그래픽이 아닐지라도 흥행에 성공하는 경우가 잦아지고 있습니다.

최근 젊은 학생들에게는 방탈출을 취미로 하는 학생들이 모인 동아리가 있을 정도로 방탈출이라는 산업이 발전하고 있습니다. 과거 많은 학생들이 pc방 당구장을 갔던 수요가 방탈출과 보드게임카페로 이동했습니다.

이러한 점에서 저희는 멀티플레이라는 개념과 탈출하는 개념을 합친 멀티플레이 전진형 퍼즐풀이 게임을 개발하는 것을 목표로 하고 있습니다.

 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	Ai Senma	
	팀 명	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

## 2 수행 내용 및 중간결과

### 2.1 계획서 상의 기획내용

#### 2.1.1 팀 운영 방식

##### 회의 주기 및 방식

매주 월, 금 총 2회 – 월요일은 비대면 디스코드로 진행하였고 금요일은 캡스톤 시간 이후로 오프라인으로 진행하였습니다.

이외 각 개발 단위별로 회의가 필요한 경우 디스코드에서 비정기적으로 회의를 진행했습니다.

##### 커뮤니케이션

메신저는 카카오톡을 기본으로 사용하였습니다. 이외 화면공유 혹은 통화가 필요한 경우 최초 줌을 사용하였으나 인원제한과 동시 화면공유의 어려움으로 인해 디스코드를 채택하여 사용하였습니다.

##### 형상관리

학교에서 제공해준 github 레포지토리를 사용하였으며 여기에 브랜치를 나누어 프로젝트를 관리하였습니다. 내부적으로 4종류의 기능으로 인원을 분배하였는데 이에 맞추어 4개의 브랜치로 나누었고 여기에 작업 후 매주 있는 회의때마다 이를 메인으로 병합하였습니다.

##### 에셋 구매 관련

게임의 개발 장르 특성상 디자이너가 있어야 그래픽 혹은 오브젝트들을 구성하고 만들 수 있습니다. 하지만 저희 팀에는 디자인 작업을 할 수 있으신분이 없으셔서 에셋을 구매하여 사용하기로 했습니다. 개발을 초기에 진행하며 2개의 에셋을 일단 구매하여 작업을 진행하기로 하였고 이에 따라 각자 대략 2만원 정도의 돈을 모아 에셋을 구매하였습니다.

##### 팀의 방향성

저희팀의 경우 모두의 이해가 일치하여 비주얼, 그래픽에 대해 이쁘게 만들기 보다는 수많은 기능들을 게임속에 최대한 많이 넣어 추후 취업에 도움이 될 수 있도록 결정했습니다.

예를 들어 싱글플레이보다는 멀티플레이로 게임을 개발하고 퍼즐 혹은 단순한 방 탈출이라면 기믹을 간단하게 만들기 보다는 캐릭터에게 여러가지 능력을 부여하여 퍼즐의 난이도와 개발적인 챌린지를 조금 더 가져가는 방식으로 개발하기로 하였습니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	<b>Ai Senma</b>	
	<b>팀 명</b>	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

## 2.1.2 게임 배경 및 스토리

### 모티브가 된 게임

- '포탈'과 '12분'이라는 게임을 참고하였음
- 실험실 배경의 '포탈' 배경과 기억을 잃고 '12분' 처럼 서사를 쌓은 다음 반전을 발견하는 내용으로 흐름을 잡음.

### 스토리 라인

- 주인공(카메라 주체)은 갑자기 놓인 새로운 환경에 당황하지만 이내 모르는 음성이 시키는대로 하기로 마음먹고 움직임.
- 흐름에 따라 초능력을 하나씩 갖게 되고 적응하기 위해 훈련식의 퍼즐풀이를 하게 됨. 마침내 주어진 퍼즐을 다 풀고 밖으로 나왔지만 주변에 남은 것은 하나도 없음. 진엔딩으로 가게 되면 이내 사건의 진상을 알게 됨

### 플레이어 엔딩 종류

- 일반엔딩 - 클리어 이후에 자신에 대한 아무 힌트로 얻지 못한 주인공으로 끝나는 엔딩
- 진엔딩 - 주인공은 사실 고아와 노숙자들을 납치하여 초능력을 부여하고 적응하는지 지켜보던 연구소(현재 배경)의 소장이었음. 납치도 한계에 다다랐고 회사 자본도 바닥을 보였기에 연구 막바지까지 끌고 온 것이 아까웠던 그는 각성을 위해 모든 기억을 지우고 직접 실험에 참가하기로 함. 실험 진행을 도와주던 Ai Senma는 자신이 연구비를 위해 신체를 팔아버렸던 친동생을 업로드한 것임. 최근에 부여된 Ai Senma라는 이름은 기억상실을 의미하는 Amnesia였던 것. 그것이 Ai로 전락한 동생의 마지막 힌트였는지, 자신이 기억을 지우기 전에 스스로에게 힌트를 주고 싶었던 것인지는 모르겠지만 이젠 유일한 혈족을 팔아버리고 이용한 자신에게 한말이 남. 그동안 비인도적으로 살았던 시간을 참회하며 영웅으로 살지, 아니면 여전히 욕심을 실현하며 살지 결정하는 순간을 암시하며 게임이 끝남. (열린 결말)

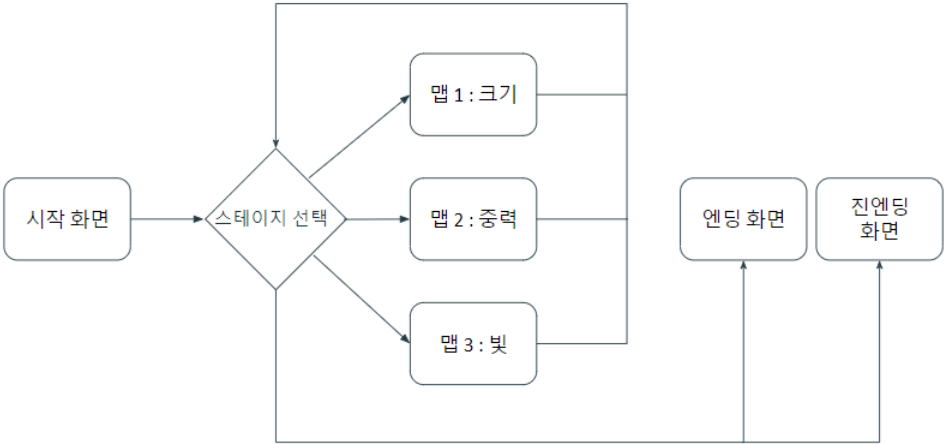
\*일반엔딩과 진엔딩의 차이가 크기에 힌트를 주는 오브젝트들을 무시하지 않고 주의깊게 살펴며 퍼즐을 푼다면 자연스럽게 진엔딩으로 흘러가게끔 할 예정

-> 진엔딩 맵으로 진입할 수 있는 힌트들


- 1 : 처음 시작할 때 주어지는 주인공의 프로필과 로딩화면에서 Ai Senma가 주는 힌트 대조
- 2 : 로딩화면에서 Ai Senma와의 대화에서 자신의 동생의 이름 힌트 획득
- 3 : 로딩화면에서 스테이지 선택 맵에 진엔딩으로 가는 숨겨진 문이 있다는 힌트 획득
- 4 : 진엔딩 맵 비밀번호인 생년월일에 대한 힌트 획득

\*로딩화면 힌트는 개발정도에 따라 채팅로그 기능을 이용하여 대화하듯 주어질 수 있음

 <div>국민대학교 소프트웨어학부 캡스톤 디자인 I</div>	중간보고서		
	프로젝트 명	Ai Senma	
	팀 명	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28



플레이어 기준 맵 시나리오 흐름도

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	<b>Ai Senma</b>	
	<b>팀 명</b>	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

### 2.1.3 카메라 및 플레이어 이동

#### 카메라 시점 종류 기획 - FPS와 TPS

아무래도 퍼즐을 직관적으로 보고 플레이어의 움직임을 보여줄 필요가 없어진다는 점에서 FPS로 기획을 하게 되면 편한 부분이 많아집니다. 다만 퍼즐 기믹의 종류를 고려했을 때 시야가 넓은 편이 아무래도 좀 더 진행상황을 파악하기 쉽고 그 퍼즐에 대한 이해도가 높아지기에 시점을 변경해서 TPS로 하게 되었습니다.


#### 카메라의 이동 방식

두가지 방안이 논의되었습니다. 먼저 첫번째 방안은 카메라를 이동할 때 마우스가 우클릭이 된 채로 카메라를 회전시키는 것과 두번째 방안은 마우스의 클릭 없이 마우스의 이동만으로 카메라를 회전시키는 방식입니다. 각각의 방법 모두 장단점이 있지만 마우스의 버튼이 늘어나게 될 경우 유저가 마우스 사용에 복잡함을 느끼게 될 것 같아 두번째 방법으로 하게 되었습니다.

#### 플레이어 이동

처음에는 마우스 클릭을 통해 플레이어를 원하는 위치로 이동시키는 방법을 생각했었습니다. 다만 그렇게 하면 게임이 너무 단조로워질 것 같아 WASD를 사용해 플레이어를 이동시키는 방법으로 하게 되었습니다. 그리고 플레이어의 전진방향은 카메라의 회전방향을 가져와서 유저가 보는 방향으로 플레이어가 이동할 수 있게끔 편리하게 구성했습니다.



 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	<b>Ai Senma</b>	
	<b>팀 명</b>	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

## 2.1.4 퍼즐 기믹 구성 – 엄력 및 크기 조정

### 단계 구성

처음 세가지의 기믹을 구성할 때 각 기믹별 스테이지마다 난이도가 올라갔으면 해서 크기/중력/빛으로 3가지의 기믹을 정했습니다.

### 엄력

플레이어가 어떤 물체를 올바르게 매칭되는 발판에 올려 단계를 나아가는 기본적인 퍼즐을 만들었습니다. 기본적인 조작을 알려주는 튜토리얼과 비슷한 역할로 구상했습니다.

### 크기 조정

플레이어가 물체의 크기를 변화시켜 넘을 수 없는 벽을 넘거나 좁은 구멍으로 물체를 통과시키는 등의 기본적인 퍼즐을 구현할 계획입니다. 이 기능도 마찬가지로 다음 퍼즐기믹을 위한 범용성을 생각해서 간단하지만 두루두루 쓰일 수 있게 구상했습니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	<b>Ai Senma</b>	
	<b>팀 명</b>	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

## 2.1.5 퍼즐 기믹 구성 - 중력

### 컨셉

중력/무중력 상태에서 이동 방법의 차이  
WASD 이동 동일 / 무중력 일 때 Q E 를 통해 위아래로 이동 가능  
실제 우주에서 움직이듯, 가스 분출을 통해 회전 및 이동 구현  
별 클릭하면 그 별의 중력에 의해 끌려가며 이동  
무중력 상태에서는 힘을 준 방향으로 멈추지 않고 이동

### 중력 퍼즐의 활용 방안

#### 이동방법을 고려한 퍼즐 - 3D 큐브 미로 탈출

- 무중력/중력 상태가 랜덤하게 바뀌며 중력과 무중력 상태에서 이동할 수 있는 공간을 구분하여 미로를 탈출하기

#### 힘(중력, 밀어내는 힘 등)의 방향을 전환하여 만드는 퍼즐

- 중력을 상속 받는 오브젝트 생성
- 중력 방향의 변환
- 오브젝트가 받는 중력의 방향을 변화시켜 해당 오브젝트의 위치, 방향을 변화시키며 퍼즐 풀이

#### 중력의 세기 조절

- 중력을 토글할 수 있다.
- 중력이 잠시 줄어들었다가 기존의 크기로 돌아오는데 중력의 세기도 같이 변하면서 오브젝트와 만났을 때 다양한 변화를 만들어낼 수 있게 한다.

#### 중력의 세기 조절 - 포격 게임 (포트리스 참고)

- 중력 등 다양한 힘을 가지고 있는 장애물 오브젝트를 이용하여 발사체의 방향을 바꾸어 원하는 곳에 도달시키기

### 이외 기획

- 슈퍼 마리오 Wii 갤럭시 어드벤처를 참고하여 행성중력을 가지는 오브젝트 제작
- 중력을 가지고 있는 오브젝트 위라면 캐릭터가 지면에 발을 딛고 서있게 한다.
- 무중력 또는 일정한 방향으로만 중력이 가해지는 상황에서 퍼즐 풀이

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	<b>Ai Senma</b>	
	<b>팀 명</b>	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28


## 2.1.6 퍼즐 기믹 구성 – 빛

### 빛 관련 기획 내용

우선 빛에 색깔을 이용해서 퍼즐을 구성하고 싶었다. 천장에 달린 빛의 색을 바꾸면 그 색에 따라 문이 생기거나 길이 생기는 형식의 퍼즐입니다. 그리고 이 때 색을 바꾸는 방법으로 플레이어가 직접 빛을 전등에 쏘서 바꾸게 하는 게 목표입니다. 그리고 빛을 오래 보면 눈에 잔상이 남는 경험이 있을 것입니다. 이를 살려 내가 빛을 계속 쏘면 물건이나, 길 또는 벽에 잔상이 남아 천장의 색을 바꿔도 안 보일 물건을 보이게 하거나, 길을 보여주게 하고, 또는 벽에 힌트를 남겨 잔상을 덧칠해야 보이는 식으로도 구성해 미로를 탈출하는 느낌을 내고 싶었습니다.

빛이 일자 형식으로 나가게 구성하다 보니 이 빛을 이용한 거울에 반사되는 퍼즐을 만들고 싶었습니다. 우선 플레이어가 쏘는 빛은 빛을 쏘게 만드는 물체에만 영향을 주게 합니다. 그리고 빛을 쏘게 만드는 물체에 도달하는 과정을 거울의 특징을 가진 블록들에 플레이어가 직접 쏘 튕겨 나가며 맞추는 방식입니다. 그 레이저를 이용하여 거울에 반사하도록 하여 특정 벽의 색상을 바꾸거나 혼합해서 퍼즐을 풀어나가는 것도 생각했습니다. 빛을 이용한 또 한가지 방법으로는 직접 레이저를 쏘 버튼을 누를 수 없는 위치인 천장 구멍 속에 배치합니다. 그리고 플레이어가 쏘는 레이저는 쏘고 1초후에 사라지는 방식으로 구현하여 한번 쏘고 나면 잔상이 남도록 합니다. 플레이어는 거울을 배치하고 쏘고 어떤 방향으로 천장 구멍속의 목표에 근접하는지 체크하면서 퍼즐을 풀어나가게 하는 것도 좋을 것 같다고 생각했습니다.

카메라는 물체에서부터 오는 빛을 통해 필름에 인화하는 방식이다. 이를 사용하여 사진을 찍고 길을 만들거나, 내가 갈 수 없는 곳의 사진을 찍어 앞에 생성하여 그 곳에 물체를 가져오는 방식으로 퍼즐을 구성하고 싶었습니다. 그곳에 힌트를 담아두거나 또는 이스터에그, 진엔딩 단서로 이용하고 싶었습니다. 이는 퍼즐 어드벤처 게임 뷰파인더에서 영감을 받아 이용하고 싶다고 생각했습니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	<b>Ai Senma</b>	
	<b>팀 명</b>	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

## 2.1.7 멀티 플레이

### 멀티플레이 관련 기획 내용

#### 컨셉

- 포탈 시리즈를 참고하여 작업함

#### 멀티플레이 코어 코드


- Unity 에서 제공하는 멀티플레이 코어인 netcode를 사용하기로 함

#### 멀티플레이 지향점

- 두명의 플레이어가 싱크되어 각자의 포지션과 상황 등을 동기화 하여 볼 수 있도록 함
- 각자의 플레이어 별로 오브젝트가 다르게 생성되지 않도록 오브젝트들을 동기화하고 이에 대한 코드 개발을 목표로 함
- 각 플레이어가 하나의 씬이 아닌 다른 씬에 있더라도 추가적인 코드를 작성하여 다른 호스트가 아닌 클라이언트가 처음 가는 공간이라도 오브젝트가 서로 동기화 될 수 있도록 마일스톤을 설정

#### 이외 기획

- Unity Gaming Services를 사용하여 대쉬보드, 매치메이킹, 로비 등을 개발

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	<b>Ai Senma</b>	
	<b>팀 명</b>	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

## 2.1.8 맵

### 맵 컨셉

- 흰 방의 연구소 -> 투명한 창문의 연구소로 변경

### 시작 맵

- 3개의 스테이지로 가기 위해 거쳐가는 정육면체의 맵
- 3개의 스테이지 클리어 시 다시 시작맵을 통해 엔딩 스테이지로 가게 됨

### 본 스테이지


- 각 스테이지의 기믹을 사용한 퍼즐을 맞춰야 다음 스테이지로 갈 수 있게 됨
- 이를 위해 크기 조정 기믹의 맵에서는 통로의 위치나 맵의 구조 변형 등을 고려함
- 중력 맵에서는 물체 뒤집기 등을 고려했고, 추가적인 변경사항 현재 계획 중

### 지향점

- 기능 및 기믹과 최대한 밀접하고 효과적으로 사용할 수 있도록 함

### 이외 기획

- 개발된 기능 및 기믹을 테스트해보기 위한 테스트 맵 개발

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	<b>Ai Senma</b>	
	<b>팀 명</b>	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

## 2.2 수행내용

### 2.2.1 프로젝트 진행 내용 - 플레이어 및 카메라 개발

#### 플레이어


- 플레이어의 움직임을 유니티에 내장된 함수인 Input.KetDown함수로 받고 그에따른 움직임을 보여줌.
- 플레이어가 움직일 시 애니메이션의 isMove 변수를 변경하여 플레이어의 애니메이션을 출력함.

#### 카메라

- 마우스의 움직임을 받아 카메라를 회전시킨다. Cinemachine이라는 에셋을 활용해서 카메라의 이동경로를 조절하고 벽등의 물체가 플레이어와 카메라 사이에 위치할 때는 카메라가 가까이 당겨지는 식으로 했다.

#### TPSCharacterController

- CharacterBody에 이동하려는 캐릭터의 몸체를 설정
- getCamera를 통해 움직이고 회전할 카메라를 설정
- Animator에 플레이어의 애니메이션을 설정
- 먼저 Awake에 플레이어컨트롤러와 Cinemachine을 설정
- Input.GetAxis를 통해 마우스의 수직 수평 움직임을 가져오고 카메라 회전

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	<b>Ai Senma</b>	
	<b>팀 명</b>	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

## 2.2.2 프로젝트 진행 내용 - 염력 및 크기 조정

### 염력

- 물체를 잡고 옮기는 기본적인 기능을 가지고 OnMouseDown/OnMouseDrag를 통해서 물체의 마우스 클릭을 입력으로 받아서 물체에 상호작용하는 코드입니다. 카메라에서 마우스의 위치를 스크린 포지션에서 받고 월드포지션으로 바꾼 후 물체를 이동시키는 역할을 합니다.

### 크기 조정

- 마우스가 물체 위에 올라갔을 때 OnMouseUp 함수를 통해 이를 인식하고, 플레이어가 매핑한 키(R과 T)를 사용하여 물체의 크기를 조정합니다. 추후에는 Input.GetMouseButton를 사용해 마우스 스크롤로도 크기 조정이 가능하도록 할 계획입니다.

### 물체 회전

- 크기 조정과 마찬가지로 OnMouseUp을 통해 물체 위에 마우스를 올림으로써 회전을 물체를 선택하여 회전 혹은 변형 할 수 있게 합니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	<b>Ai Senma</b>	
	<b>팀 명</b>	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

## 2.2.3 프로젝트 진행 내용 - 중력

### 작업 설명 - 중력

- 무중력을 기본 배경으로 하고 특정 공간에서 중력의 방향을 정하는 GravityArea와 그 방향을 향해 힘을 가해주는 GravityBody를 중력에 관련된 모든 기믹에서 사용합니다. 중력의 방향은 GravityArea에서 알 수 있고 이를 컴포넌트로 하는 오브젝트에 대한 Collider 충돌에서 시작합니다. GravityArea의 공간으로 다른 Collider가 충돌한다면 OnTriggerEnter/Exit의 판단을 통해 List<GravityArea>에 추가 또는 삭제되게 됩니다. 이때 충돌한 오브젝트의 컴포넌트로 있는 GravityBody는 List<GravityArea>의 우선순위를 확인하여 가장 큰 우선순위 값을 가진 GravityArea의 중력 방향 값을 받아와 저장 후 Rigidbody에 그 방향에 대한 AddForce를 실행하여 중력을 제공하였습니다. 이를 통해 특정 공간에서 원하는 방향의 중력을 생성할 수 있습니다.
- 플레이어가 가지고 있는 기능은 두가지로 RayCast를 이용하여 충돌 여부를 확인하는 것을 기본으로 합니다. 충돌한 오브젝트의 태그가 'GravityObject', 'moveable'일 때의 기능을 달리합니다. 'GravityObject' 태그의 오브젝트와 충돌한 경우 GravityHandler 스크립트의 ToggleGravity함수를 호출합니다. ToggleGravity함수는 RayCast Hit한 'GravityObject' 태그 오브젝트의 자식으로 'GravityArea' 이름의 자식이 있는지 확인하고 있다면 GravityArea Scale에 대한 애니메이션을 적용한 Coroutine을 수행합니다. 이는 약 2초간 수행되며 GravityArea의 Scale을 줄였다 다시 커지게 하는 애니메이션을 제공합니다. 이를 통해 중력에 영향을 받는 오브젝트는 줄어들고 커질 때의 GravityForce 크기의 변화를 받으며 다양한 방향으로 이동하게 되고 이는 게임의 변칙성을 주어 난이도를 조절할 수 있게하기 위해 생성되었습니다.
- 중력 퍼즐의 게임 요소 중 하나로 오브젝트를 중력 또는 다양한 힘의 영향을 받게하여 원하는 공간에 도달시키기가 있습니다. 만약 도달에 실패하면 무중력 상태에서 여러 힘의 영향을 받은 오브젝트는 멈추지 않고 계속해서 변칙적으로 움직이게 됩니다. 하지만 도달이 목적이고 성공의 여부이기 때문에 실패해도 여러번 다른 방식으로 시도할 수 있게끔 하기 위해 오브젝트를 초기위치로 이동시키는 기능이 추가했습니다. 이는 'moveable'태그를 가진 오브젝트가 RayCast에 Hit되었을 때 실행됩니다. 초기 Position으로 되돌리면서 계속되는 움직임을 제어하기 위해 Rotation을 0,0,0으로 초기화하고 Rigidbody의 isKinematic을 Off->On 하여 물리적 작용을 없애 초기 상태의 오브젝트로 되돌릴 수 있습니다.



 <div> <b>국민대학교</b>  <b>소프트웨어학부</b>  <b>캡스톤 디자인 I</b> </div>	중간보고서		
	프로젝트 명	Ai Senma	
	팀 명	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

## 코드 관련 설명 - 중력

- GravityArea
  - SerializedField를 통해 우선순위 설정
  - 중력의 방향을 결정하는 GravityArea 가져오기
    - Collider의 충돌을 통해 OnTriggerEnter/OnTriggerExit 확인하여 GravityArea를 List<GravityArea>에 추가/삭제
    - GravityArea를 상속받은 Gravity Up/Center/Point 등의 클래스는 중력의 방향을 정의하는 메서드가 구현되어 있음
- GravityBody
  - GravityArea 타입을 저장할 List 생성
  - AddGravityArea 함수 : GravityArea를 List에 추가
  - RemoveGravityArea 함수 : GravityArea를 List에서 삭제
  - 중력 방향을 설정하기
    - List 내의 가장 큰 우선순위 GravityArea의 방향값을 GetGravityDirection 함수를 통해 호출
  - 받아온 방향값으로 Rigidbody에 AddForce
  - 중력 방향에 맞게 캐릭터가 서있을 수 있게 회전 조절
- GravityController
  - 마우스 포인트가 있는 곳에 RayCast로 충돌여부 확인하기
    - 'GravityObject'라는 태그를 가진 오브젝트와 충돌한 경우
      - GravityHandler 스크립트의 ToggleGravity 함수 호출하기
    - 'moveable'이라는 태그를 가진 오브젝트와 충돌한 경우
      - 해당 오브젝트의 초기위치, 회전 설정하기
- GravityHandler
  - ToggleGravity 함수
    - 자식 RayCast Hit한 'GravityObject' 태그의 오브젝트의 'GravityArea' 이름의 자식이 있는지 확인하기
      - 있다면 GravityArea Scale에 대한 애니메이션을 적용한 Coroutine 수행

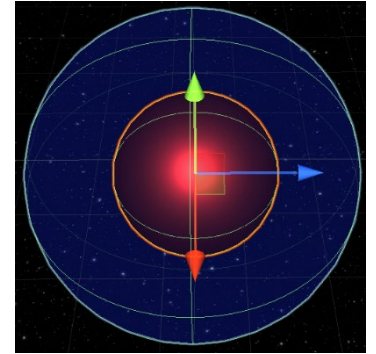


그림 1 파란색이 GravityArea

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	<b>Ai Senma</b>	
	<b>팀 명</b>	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28


## 2.2.4 프로젝트 진행 내용 - 빛

우선 빛을 쏘기 위해서 raycasthit 함수에 대한 선행이 필요했습니다. 그렇게 일단 큐브 오브젝트를 이용하여 raycasthit을 쏘 물체와 충돌하거나, 충돌하지 않을 때 일정거리만큼 나아간 위치를 제대로 받아오는지 검증했다. 이때 필요한 것은 물체의 위치와 방향을 이용했습니다.

그 이후 내가 원하는 빛을 쏘는 방식은 거울에 부딪혔을 때 튕겨나가는 방식입니다. 따라서 도착한 지점의 tag가 mirror라면 direction을 90도로 변경하는 방식으로 구성해 나아가고 그 도착지점에 대해 검증했습니다. 이 때 mirror가 아니라 일반 물체에 충돌하면 그냥 빛은 멈추도록 구성했습니다.

이제 내 빛이 잘 나아가고 충돌했을 때는 때에 따라 반사되거나 진행 방향이 멈추도록 설계를 완료했으니 이제 Rendering을 준비했다. 우선 외형을 구성할 material이나 나는 색을 이용할 것이기 때문에 색깔의 요소도 light를 생성하는데 필요하다. 그 밖에 선의 굵기와 같은 요소들도 추가하여 만들어준다. 이 만들어진 요소를 이용해 충돌하거나 충돌하지 않은 raycasthit을 통해 모든 좌표들로 빛의 렌더링 작업을 거친다.

이제 플레이어가 빛을 쏘거나 물체가 빛을 쏘는 방식은 완료했고 이제 벽에 색을 입히고 싶었습니다. 우선 내가 쏘는 빛은 벽에 영향이 없게 하고 포인터가 쏘는 빛에만 영향이 가게 했습니다. 이는 추후 수정될 수도 있습니다. 포인터에서 나온 빛을 거울을 이용해 벽에 닿게하면, 우선 벽인지 검증하고 벽에 쏜 한가지의 빛의 색깔을 벽에 담아두고 벽의 색을 업데이트하게 했습니다. 한 가지 색상의 테스트를 완료한 후 두 가지 색상의 빛을 동시에 맞춘다면 섞이게 구현했습니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
	프로젝트 명	Ai Senma	
	팀 명	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

## 2.2.5 프로젝트 진행 내용 - 네트워크 동기화 기능

### 네트워크 관련 기초 작업

- 호스트 포트 열기, 게스트가 호스트 접속하기: 필요한 NetworkObject가 생성될 때까지 기다린 후 worldScene으로 이동한다.
- 서버에서 NetworkObject 스폰함
- NetworkPrefabList 설정함
- NetworkVariable, RPC 기능 사용함

### 네트워크 관련 편의성 개발

- MultiplayerBuildAndRun: 최대 4개의 빌드를 쉽게 실행시켜준다.
- ConsoleToGUI: 빌드에서도 DebugMessage를 보게 해준다.

### 멀티플레이 관련 세부 개발

- MyNetworkTransform, NetworkSceneManager 개발함
- NetworkVariable 사용해 객체 Transform 동기화함
- Owner의 경우 write: 자신의 TransformData를 전파함
- sendPerSec: 초당 데이터를 몇 번 보낼 지 설정할 수 있게 함

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	<b>Ai Senma</b>	
	<b>팀 명</b>	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

### 멀티 Scene 관련 세부 기능 개발

- bKeepSendWhenNotExist: LoadScene을 통해 다른 Scene의 위치 데이터를 보내게 함
- Owner가 아닌 경우 read: TransformData를 보간하여 위치를 업데이트함
- 다른 Scene에 있거나 SceneSequenceNumber가 일치하지 않을 때는 무시하기
- 기존 Scene에서 나갈 때: 자신이 기존 Scene에서 나갔음을 알린다: NotifyLeaveSceneRpc()
- 시야에서 사라지기: 모든 MyNetworkTransform에 대해 SetExist(false) 메세지 Invoke()하기

### 신규 Scene 진입시 오브젝트 처리 기능 개발

- 시야에서 나타나기: 내가 owner이고 새로운 scene에 있는 MyNetworkTransform에 대해 SetExist(true) 메세지 Invoke()하기
- 해당 Scene에 이미 입장한 NetworkObject의 정보를 요청하면서, 자신이 새로운 Scene에 들어갔음을 알린다: NotifyReadyForPlacedRpc(ReadyForPlacedType.MoveScene)
- 새로운 Scene에 입장 직전, 모든 NetworkTransform의 m\_isReadyForPlaced와 m\_isPlaced를 false로 만든다.
- NotifyReadyForPlacedRpc(ReadyForPlacedType.EchoBack)이 도착하면 m\_isReadyForPlaced를 true로 만든다: FixedOnPlace
- TransformData 패킷이 도착했을 때, m\_isReadyForPlaced가 true이면 보간하지 않고 바로 배치시킨 후 m\_isPlaced를 true로 만든다.
- m\_isPlaced가 true일 때는 평소대로 보간한다.


 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
	프로젝트 명	Ai Senma	
	팀 명	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

## 게임 내부 오브젝트 객체 스폰 및 동기화 기능 개발

- static RegisterMap 관리: 이미 Spawn된 객체인지 확인하기
- IsPlacedByDesigner가 true이고 RegisterId가 등록되지 않았으면 서버에 Spawn 요청하기: SpawnIfNotSpawnedRpc()
- 이미 Spawn요청된 RegisterId이면 무시하기
- ClientSideSpawning: 호스트가 아직 가보지 않은 Scene에 클라이언트가 최초 입장하더라도 정상적으로 네트워크에서 Spawn된다.
- SendTo.Server를 사용하여 클라와 서버가 동일한 로직으로 객체를 생성한다.
- SpawnIfNotSpawnedRpc() 호출 후 자신을 삭제
- IsPlacedByDesigner가 false인 경우 Register하고 Spawn된 NetworkObject이므로 삭제하지 않는다.
- SpawnInfo를 통해 CurrentSceneIndex와 SceneSequenceNumber를 확인함

## 이외 기능 개발

- owner이면 OnNetworkSpawn()에서 NotifyReadyForPlacedRpc(ReadyForPlacedType.EchoBack)
- RPC가 OnNetworkSpawn()이 호출되기 전에 도착한 경우 fault가 만나게 수정: MyNetworkTransform이 null값일 수 있는 SpawnInfo 구조체 개발
- MoveScene: NetworkTransform과 함께 새로운 Scene으로 이동한다. UnityAction인 SetSpawnPositionEvent()를 Invoke하여 새로운 Scene에서의 최초 위치를 결정하게 한다.
- LoadScene: 마치 유체이탈처럼 관전자 입장으로 다른 Scene을 볼 수 있다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	<b>Ai Senma</b>	
	<b>팀 명</b>	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

### 3 수정된 개발 내용 및 개발 과정

#### 3.1 수정사항 및 문제해결 과정

##### 3.1.1 플레이어 및 카메라 개발

**문제:** 카메라 이동방식을 마우스 오른쪽 버튼 클릭과 마우스 이동을 동시에 해서 시점을 돌릴지 마우스 이동으로만 회전을 할 지 정하는 방식에서 시행착오를 많이 겪었음.

**해결방법:** 결과적으로는 마우스의 버튼 사용 증가가 플레이어에게 혼란을 야기할수 있다는 걱정에 마우스의 오른쪽버튼 사용을 없애고 마우스 이동으로만 카메라를 회전시키기로 함

**문제:** 카메라와 플레이어 사이에 물체가 위치할 시 카메라가 밖으로 나가버리는 현상이 있었음

**해결방법:** 이를 유니티 에셋인 cinemachine을 이용해서 카메라 자체에 콜라이더를 부여하는 방식으로 해결함

##### 3.1.2 염력 및 크기 조정

**문제:** 기능의 개발은 따로 어려운 사항은 없었고 실제 마우스의위치와 움직인 물체의 위치가 멀어질수록 오차가 생기게됨. 따라서 아예 마우스 위치로 정확히 이동을 시키는 방법으로 변경함

**해결방법:** RayCast를 사용하여 오브젝트를 이동하고 수정할 때 시야에 따라서 오차가 발생하였음, 이 방식은 추후에 버그 및 유지보수성이 떨어진다고 판단하여 플레이어의 카메라에 오브젝트를 상대위치로 고정하여 이동하는 방식으로 개선하였음

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	<b>Ai Senma</b>	
	<b>팀 명</b>	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

### 3.1.3 중력

**문제:** 유니티 physics.gravity의 x,y,z 축의 값의 변경을 통해 중력의 방향을 임의로 지정하는 부분에서 맵에 있는 물리적인 상호작용을 하는 모든 오브젝트에 영향을 주기 때문에 모든 게임 과정을 physics.gravity가 변화했을 때를 고려하고 개발해야하기 때문에 불편함을 초래함.

**해결방법:** 이를 해결하기 위해 오브젝트가 위치한 곳에 일정한 방향의 중력 역할을 하는 스크립트로 중력 구현 변경함.

**문제:** GravityArea가 겹쳤을 때 어떤 방향의 중력을 받아야하는지에 대한 고민이 있었음.

**해결방법:** 오브젝트 마다 GravityArea 우선순위를 적용하여 겹쳤을 때 선택된 방향의 중력을 받게끔 하여 문제를 해결하였음.

**문제:** 모든 중력의 힘이 항상 같기에 그 영향을 받은 오브젝트가 항상 같은 운동을 하는 문제가 발생함.

**해결방법:** 퍼즐게임에서 중력을 토글하여 작고 크게 함으로써, GravityForce의 변화를 통해 오브젝트의 움직임에 변화를 주고 싶었기에 GravityScale에 따른 중력의 크기에 변화를 주었다.

**문제:** 중력장(GravityArea) 안에(겹쳐) 있는 특정 오브젝트는 중력장에 막혀 RayCastHit이 되지 않는 문제

**해결방법:** RayCast는 GravityArea라는 Layer를 통과하게끔 변화를 주고 기존 방법은 GravityArea에 RayCastHit이 중력에 대한 토글 방법이었지만 중력을 가지고 있는 오브젝트에 RayCastHit이 되었을 때로 변경

#### 이외 문제들

중력의 방향이 변할 때 캐릭터 Rigidbody의 변화에 의해 캐릭터는 항상 지면을 90도로 서기 때문에 캐릭터가 점프해야하는 방향값을 계속 갱신해주었음

중력 기믹의 기능적인 요소들은 네트워크에 대한 고려 없이 완성된 것으로 멀티플레이, 네트워크 환경에 맞추어 많은 변경이 발생할 예정

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	<b>Ai Senma</b>	
	<b>팀 명</b>	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

### 3.1.4 빛

**문제:** 빛을 쏘고 거울에서 반사될 때 그 반사되는 지점이 물체 안에 파묻히는 버그가 발생하였음

**해결방식:** 버그가 발생한 이유를 찾고 있으며 곧 수정될 예정임

**문제:** 레이저가 여러 개가 합쳐지며 색이 섞여야 하지만 섞이지 않는 문제가 있었음

**해결방식:** 벽에 스크립트를 추가해 색상을 담아두는 기능을 추가하며 해결함

### 이외 문제들

현재는 네트워크와 병합하여 멀티플레이에 대한 작업을 하고 있는데 이는 혼자 게임하는 것으로 구성된 나의 코드들과 차이가 있었다. 현재 병합 초기이나, 다양한 부분들이 바뀔 것 같다.

### 3.1.5 네트워크 개발

**문제:** 게스트가 호스트가 가보지 못한 썬에 도달할 경우 오브젝트를 로딩하지 못하거나 동기화하지 못하는 문제가 발생함

**해결 방식:** 이를 해결하기 위해 MyNetworkTranform과 Register List를 구현하여 RPC 및 여러가지 오브젝트들을 호스트가 확인후 스폰 및 관리하도록 하였음

**문제:** 호스트가 만든 방에 게스트가 진입할 때 이미 생성된 오브젝트를 중복으로 생성할 우려가 있음

**해결방식:** 클라이언트에서 가지고 온 오브젝트를 호스트에서 중복으로 처리하지 않도록 모든 오브젝트를 RPC로 요청할 때 Register List에서 확인후 오브젝트를 구성하도록 구현하여 이를 해결하였음



 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	<b>Ai Senma</b>	
	<b>팀 명</b>	LAB 36	
	Confidential Restricted	Version 1.0	2024-03-28

## 4 향후 추진계획

### 4.1 향후 계획의 세부 내용

#### 프로젝트 전반에 걸친 문제와 향후 계획

- 프로젝트 전반에 걸쳐 우리의 프로젝트에는 큰 문제가 하나 존재합니다.
- 네트워크 개발에 대한 초기 배경지식의 부재로 각각의 기능들이 네트워크에서 멀티플레이를 할 것을 고려하지 못한채 순수 '기능'만이 개발되었습니다.
- 이는 싱글플레이에서는 작동성이 보장되지만 멀티플레이에서의 작동성이 전혀 보장되지 않은 상태입니다.
- 위와 같은 문제로 인해 각각의 기믹을 개발한 팀원들이 네트워크를 담당한 팀원에게 각 기믹이 어떻게 네트워크에 적용되어야 하고 이에 따라 프로그램들이 멀티플레이 환경에서 잘 작동하도록 개발할 예정입니다.
- 이에 대해 우리는 중간보고 이후 2주 뒤인 4월 19일까지를 해당 기믹들과 네트워크 내부의 문제를 해결하기로 하였습니다.
- 또한 이후 퍼즐에 대한 아이디어와 회의를 통해 구체적으로 맵을 구성하고 비주얼을 향상시켜 최종발표에 임할 예정입니다.

#### 스토리 관련 향후 계획

- 진엔딩 맵 진입 도어락 기능 개발
- 로딩화면에서 대화처럼 힌트 주는 연출 기능 개발
- Ai Senma에서 Amnesia로 바뀌며 스토리 힌트 회수하는 애니메이션 개발

#### 이외 향후 계획

- 맵 틈새에 각 오브젝트의 크기를 조절하여 맞추면 열쇠처럼 돌아가는 맵 개발
- 행성중력을 부여하여 우주(무중력 상태)에서 갈고리로 이동하듯 움직이는 방법 개발
- 조명 각도를 조절하여 조명 위치에 잔상을 남겨서 맞추는 퍼즐 개발
- 네트워크 내부에서 오브젝트의 클라이언트가 owner 관련하여 권한을 이양받거나 이를 통하지 않고도 오브젝트를 움직일 수 있도록 하는 기능 개발