

# 캡스톤 디자인 I 종합설계 프로젝트

프로젝트 명	AI SENMA
팀 명	LAB 36
문서 제목	기말보고서

Version	1.1
Date	2024-05-10

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

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	AI SENMA	
	팀 명	LAB 36	
	Confidential Restricted	Version 1.1	2024-05-24


CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 캡스톤 디자인 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	문민철,강기범,강동우,김현준,박세호,이재원,오정규
수정작업자	문민철,강기범,강동우,김현준,박세호,이재원,오정규

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2024-03-28	오정규	1.0	신규작성	신규작성
2024-05-10	오정규	1.1	기말보고	기말보고 내용 추가


	<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 1.1	2024-05-24

목 차

1	프로젝트 목표 .....	5
2	수행 내용 및 중간결과 .....	6
2.1	계획서 상의 기획내용 .....	6
	팀 운영 방식 .....	6
	게임 배경 및 스토리 .....	7
	카메라 및 플레이어 이동 .....	9
	퍼즐 기믹 구성 - 염력 및 크기 조정 .....	10
	퍼즐 기믹 구성 - 중력 .....	11
	퍼즐 기믹 구성 - 빛 .....	12
	멀티 플레이 .....	13
	맵 .....	14
2.2	수행내용 .....	15
	프로젝트 진행 내용 - 메인화면 및 UI 관련 개발 .....	15
	프로젝트 진행 내용 - 전반적인 게임 수정 .....	16
	프로젝트 진행 내용 - 플레이어 및 카메라 개발 .....	17
	프로젝트 진행 내용 - 염력 및 크기 조정 .....	18
	프로젝트 진행 내용 - 중력 .....	19
	프로젝트 진행 내용 - 암흑물질 .....	22
	프로젝트 진행 내용 - 네트워크 동기화 기능 .....	24
	프로젝트 진행 내용 - 스토리 .....	27
3	수정된 개발 내용 및 개발 과정 .....	29
3.1	문제해결 과정 .....	29
	플레이어 및 카메라 개발 .....	29
	염력 및 크기 조정 .....	30
	○중력 .....	31
	빛(암흑물질) .....	33
	스토리 .....	34
	네트워크 개발 .....	35
4	중간보고 피드백 결과 .....	36
4.1	중간보고 요약 .....	36
	강점 .....	36
	부족한 점 .....	36
	조언 .....	36
4.2	중간보고 피드백 이행 .....	37

	<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 1.1	2024-05-24

	재미라는 요소에 대한 정형화 문제.....	37
	유저 피드백 정리 및 개선 사항.....	38
4.3	내부 피드백 사항.....	41
4.4	오픈 마켓 등록 .....	42
	STEAM 등록 현황.....	42
	스토브 스토어 등록 현황 .....	44
5	향후 추진계획 .....	45
5.1	향후 계획의 세부 내용 .....	45

 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	AI SENMA	
	팀 명	LAB 36	
	Confidential Restricted	Version 1.1	2024-05-24


## 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 1.1	2024-05-24

## 2 수행 내용 및 중간결과

### 2.1 계획서 상의 기획내용

#### 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. 후반부엔 월요일 회의도 대면으로 진행하며 회의록을 작성했고 필요시 다른 요일도 회의했습니다.

##### 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>	중간보고서		
		프로젝트 명	AI SENMA	
		팀 명	LAB 36	
		Confidential Restricted	Version 1.1	2024-05-24

## 게임 배경 및 스토리

### 모티브가 된 게임

- '포탈'과 '12분'이라는 게임을 참고하였음
- 연구소/실험실 배경의 '포탈' 배경을 참고함. '12분'처럼 기억을 잃고 서사를 쌓은 다음 반전을 발견하는 흐름으로 진행함.

### 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. 퍼즐을 풀고 정보를 얻으며 조건을 충족하여 진엔딩으로 가게 되면 이내 사건의 진상을 알게 됨. 만약 그러지 못할 경우 일반 엔딩인 엔딩 크래딧을 보게 됨.

### 플레이어 엔딩 종류

- 일반엔딩 - 퍼즐을 풀고 AI SENMA와 대화하며 자신에 대해 정보를 얻지만 주인공 스스로가 결국 정확히 어떤 사람이었는지는 알지 못한 채로 끝남
- 진엔딩 - 주인공은 사실 고아와 노숙자들을 납치하여 초능력을 부여하고 적응하는지 지켜보던 연구소(게임 배경)의 소장이었음. 회사 자본도 바닥을 보였고 납치도 한계라서 연구 막바지까지 진행한 것이 아까웠던 그는 각성을 위해 모든 기억을 지우고 직접 실험에 참가하기로 했던 것임. 실험을 돕는 AI는 매번 실험체의 특징에 따라 이름이 바뀌는데 이번 AI의 이름이 AI SENMA인 이유는 주인공의 의도된 기억상실을 의미하는 AMNESIA를 거꾸로 한 것. 실험 진행을 도와주던 AI SENMA는 주인공이 연구비를 위해 신체를 팔아버렸던 친동생을 정신만 AI로 업로드한 것임. 유일한 혈족을 팔아버리고 이용하며 납치와 실험 등 여러 범죄를 일으킨 자신을 알게 되고 충격을 받는 엔딩임.


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

- 진엔딩 맵으로 진입할 수 있는 힌트들

1 : AI SENMA와의 대화에서 나오는 자신에 대한 정보들, 연구소의 배경들

2 : 맵에 숨겨져 있는 사원증

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

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
		프로젝트 명	AI SENMA	
		팀 명	LAB 36	
		Confidential Restricted	Version 1.1	2024-05-24

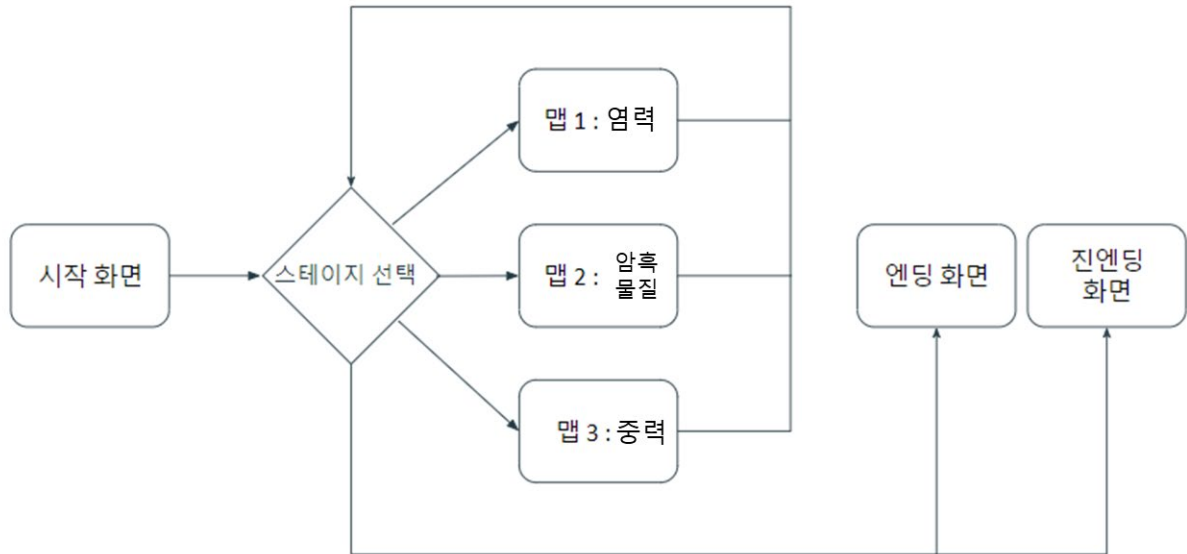


그림 1 플레이어 기준 맵 시나리오 흐름도

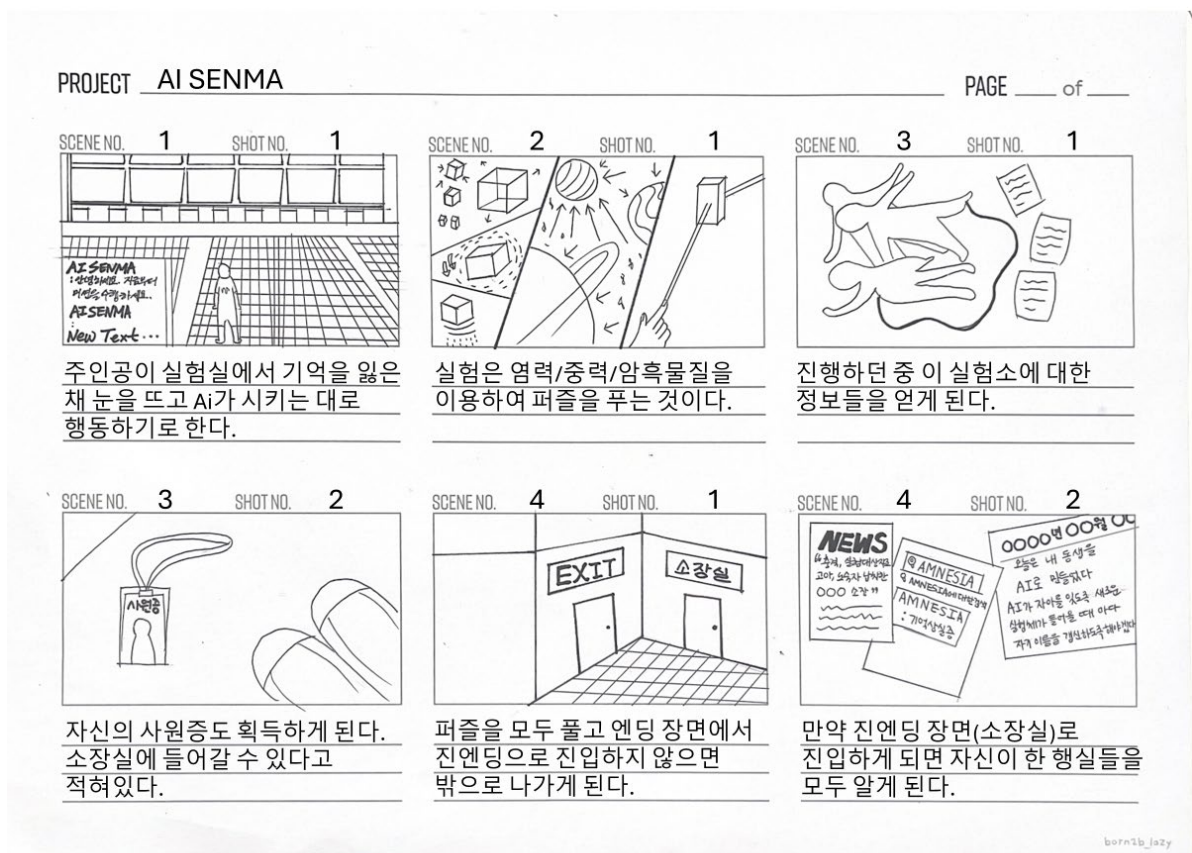



그림 2 AI SENMA 스토리 보드



 <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 1.1	2024-05-24

## 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 늘어나기에 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.

 <div>국민대학교 소프트웨어학부 캡스톤 디자인 I</div>	중간보고서		
	프로젝트 명	AI SENMA	
	팀 명	LAB 36	
	Confidential Restricted	Version 1.1	2024-05-24

퍼즐 기믹 구성 - 염력 및 크기 조정

단계 구성


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

염력

염력이라는것은 물체를 옮기는 능력이라는 것에 착안해서 물체를 던지거나 특정 구간에 옮기는 등의 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 1.1	2024-05-24

## 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>	중간보고서		
		프로젝트 명	AI SENMA	
		팀 명	LAB 36	
		Confidential Restricted	Version 1.1	2024-05-24

## 퍼즐 기믹 구성 - 빛

### 빛 관련 기획 내용

#### 퍼즐 구성

- 빛의 색깔을 이용한 퍼즐
  - 천장에 달린 빛의 색을 바꾸면 문이나 길이 생김.
  - 플레이어가 직접 빛을 전등에 쏘서 색을 변경.
- 잔상 효과 활용
  - 빛을 오래 쏘면 물건, 길, 벽에 잔상이 남음.
  - 천장의 색을 바꿔도 보이지 않는 물건을 잔상으로 보이게 함.
  - 잔상을 덧칠해야 보이는 힌트를 벽에 남겨 미로 탈출 느낌 제공.

#### 거울을 이용한 퍼즐


- 빛의 반사 활용
  - 플레이어가 쏘는 빛은 빛을 쏘게 만드는 물체에만 영향.
  - 거울 블록을 이용해 빛을 반사시켜 퍼즐 해결.
  - 특정 벽의 색상을 바꾸거나 혼합하여 퍼즐 풀이.

#### 레이저 퍼즐

- 레이저를 쏘서 버튼 누르기
  - 천장 구멍 속에 배치된 버튼에 레이저 쏘기.
  - 레이저는 쏘고 1초 후 사라져 잔상이 남음.
  - 거울을 배치하고 레이저를 쏘아 퍼즐 해결.

#### 카메라 활용 퍼즐

- 사진을 이용한 퍼즐
  - 사진을 찍어 길을 만들거나 갈 수 없는 곳을 찍어 앞에 생성.
  - 힌트나 이스터에그, 진엔딩 단서로 활용.
  - 퍼즐 어드벤처 게임 뷰파인더에서 영감을 받아 구성.

	<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 1.1	2024-05-24

## 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 1.1	2024-05-24

## 맵

### 맵 컨셉

- 흰 방의 연구소 -> 미래 배경의 연구소로 변경

### 시작 맵

- 3개의 스테이지로 가기 위해 거쳐가는 정육면체의 맵
- 3개의 스테이지 클리어 시 다시 시작 맵을 통해 클리어 방식에 따라 2가지의 서로 다른 엔딩 스테이지로 가게 됨

### 본 스테이지

- 각 스테이지의 기믹을 사용한 퍼즐을 맞춰야 다음 스테이지로 갈 수 있게 됨
- 한 스테이지 내부에서도 한 단계를 클리어해야 다음 단계로 넘어갈 수 있는 여러 단계가 존재함
- 이를 위해 크기 조정 및 염력 맵에서는 물체의 크기를 조정하지 않으면 원하는 곳으로 물체를 놓을 수 없거나 다음 목표 지점으로 가기 위한 통로를 생성할 수 없게 만들어 다음 스테이지로 넘어갈 수 없도록 설계함
- 빛 맵에서는 빨강, 초록, 파랑, 검정 빛을 쏘는 4가지의 레이저 포인터와 거울 등을 이용하여 두뇌 회전을 통해 퍼즐을 클리어하도록 설계함
- 중력맵에는 중력을 토글 할 수 있는 구체를 통해 물체를 바이패스 시켜 원하는 위치에 보내는 기능과 물체를 빨아들이는 강력한 중력을 가진 블랙홀 구체를 피해 게임을 진행 할 수 있도록 함
- 최종 스테이지인 중력 맵 후반부에 다른 엔딩으로 갈 수 있는 키(사원증)를 추가

### 지향점

- 각 맵의 기능 및 기믹을 효과적으로 사용할 수 있도록 함
- 각 스테이지의 각 단계마다 별개의 기능을 사용하기보단 이전 단계에서 사용했던 기능을 다음단계에서도 사용할 수 있도록 설계

### 이외 기획

- 개발된 기능 및 기믹을 테스트해보기 위한 테스트 맵 개발
- 블랙홀, 입자 등 각종 파티클 시스템 추가
- 각 기믹 맵 시작 지점에 시작 맵으로 다시 돌아갈 수 있는 텔레포트 생성
- 맵의 틀을 구성하기 위한 실험실 에셋 외의 박스, 빛 맵의 거울 등 디테일을 위한 여러가지
- 무료 에셋 사용

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
		프로젝트 명	AI SENMA	
		팀 명	LAB 36	
		Confidential Restricted	Version 1.1	2024-05-24

## 2.2 수행내용

### 프로젝트 진행 내용 – 메인화면 및 UI 관련 개발

#### 메인화면

- 메인화면 제작

TitleScreen.cs을 통해 버튼의 클릭을 통해 각각의 알맞은 화면으로 이동하게 함


- 싱글플레이와 멀티플레이 분리

#### UI 관련 개발

- 게임의 옵션설정 화면 제작 슬라이더의 조절을 통해 직접적으로 옵션을 설정할 수 있습니다.

#### OptionSettings.cs

- SetVolume / setBrightness / setMouseSpeed 함수를 만들어서 그 함수들을 통해 마스터 볼륨, 카메라의 밝기, 마우스의 속도를 조절
- 게임의 소리, 밝기, 마우스 속도를 조절하는 스크립트 개발

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
		프로젝트 명	AI SENMA	
		팀 명	LAB 36	
		Confidential Restricted	Version 1.1	2024-05-24

## 프로젝트 진행 내용 – 전반적인 게임 수정

### 전반적인 게임 수정

- 플레이어 이동시 및 퍼즐 클리어시의 사운드 추가
- 엔딩이 2가지이므로 그에 알맞은 엔딩 맵 추가 및 디자인

### 저장 기능 구현

- 스테이지 클리어시 JSON 파일로 저장하는 기능 구현

**방법 1:** 스테이지 입장 시 전 스테이지 클리어 여부 확인 후 입장

**방법 2:** 멀티플레이의 경우, 호스트와 게스트가 같은 스테이지 문 앞에 있을 때 입장 가능하도록 수정

**방법 3:** 호스트와 게스트가 지정된 Scene으로 동시 입장하는 로직 구현

### 퍼즐 완성

- 퍼즐을 풀었을 때 사운드를 출력하여 퍼즐의 완료를 알림

### 이외 사소한 수정

- Cinemachine의 카메라이동이 어지러운 요소가 있어 일반적인 카메라로 수정

방법 : 카메라에 콜라이더를 부여하는 부분은 RayCast를 이용해서 대체함

- 캐릭터가 무한히 점프할 수 없도록 수정

방법 : 캐릭터 바닥 부분에 BoxCasting을 통해 isOnGround 확인



	<b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
		프로젝트 명	AI SENMA	
		팀 명	LAB 36	
		Confidential Restricted	Version 1.1	2024-05-24

## 프로젝트 진행 내용 - 플레이어 및 카메라 개발

### 플레이어


- 플레이어의 움직임을 유니티에 내장된 함수인 Input.GetKeyDown함수로 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 했음.

### 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

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	AI SENMA	
	팀 명	LAB 36	
	Confidential Restricted	Version 1.1	2024-05-24

## 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.
- 염력 사용시 물체 최대속도 및 떨림 개선하였습니다.
- 물체를 너무 빨리 던져서 맵 밖으로 나가는 것을 방지했습니다.

### 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.
- 물체가 충돌을 감지하면 크기변화 불가능하게 수정 하여 버그를 방지했습니다.

### 물체 회전

- 크기 조정과 마찬가지로 OnMouseUp을 통해 물체 위에 마우스를 올림으로써 회전할 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 1.1	2024-05-24

## 프로젝트 진행 내용 - 중력

### 중력

- 중력의 기본은 Rigidbody 의 Addforce funciton 으로 다양한 방향으로 힘을 작용하는 것.
- 중력 맵에는 사용되는 오브젝트는 중력을 가지는 오브젝트, 중력의 영향을 받는 오브젝트로 나뉘는데 플레이어는 두 종류의 오브젝트와 상호작용을 통해 퍼즐을 해결할 수 있음.

### 중력 맵 설명

- 중력맵은 무중력이 배경
- 플레이어만 중력의 영향을 받음.
- 오브젝트의 Rigidbody 를 통해 중력의 유무 결정
- Rigidbody 에 특정 방향으로의 중력(힘)을 적용하는 GravityBody 스크립트 생성
- GravityBody 에 중력의 방향을 전달하는 GravityArea 스크립트 생성

### GravityBody, GravityArea 설명


- GravityArea 를 스크립트로 가지는 오브젝트는 그 범위 내에 GravityBody 와 Rigidbody 를 가진 모든 오브젝트에 GravityArea(Center, up, down, point 등 어떤 방향이든 설정 가능)가 설정된 방향으로 힘을 작용시킴
- GravityArea 는 우선순위를 가짐. 이는 중력의 크기를 사용자가 임의로 설정하여 더 큰 중력을 가진 오브젝트로 표현할 수 있게 함. 우선순위가 클수록 더 강한 중력을 가지는데 만약 오브젝트가 우선순위가 100 인 GravityArea 에 있을 때 100 보다 작은 수의 우선순위를 갖는 GravityArea 의 영향은 받지 않음. 이를 통해 오브젝트가 GravityArea 의 영향을 받는 순서를 결정할 수 있음.

### 사용한 오브젝트

- 중력을 가진 행성 오브젝트, 중력장, 오브젝트의 이동 Area, 큐브 보관 등
- 큐브 제출 Area(기믹 공통 사용)
  - 큐브가 특정 공간에 초 이상 머물러 있을 시 제출이 되는 큐브 제출 스크립트를 가진 오브젝트

### Raycast 사용 기능

- 플레이어가 가지고 있는 기능으로RayCast를 이용하여 충돌 여부를 확인하는 것이 기본. 충돌한 오브젝트의 태그가 'GravityObject', 'moveable'일 때의 기능을 달리함.

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
		프로젝트 명	AI SENMA	
		팀 명	LAB 36	
		Confidential Restricted	Version 1.1	2024-05-24

### GravityArea의 동작 방식

- GravityBody를 컴포넌트로 하는 오브젝트에 대한 Collider 충돌에서 시작. GravityArea의 공간으로 다른 Collider가 충돌한다면 OnTriggerEnter/Exit의 판단을 통해 List<GravityArea>에 추가 또는 삭제되게 됨. 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 제공. 이를 통해 특정 공간에서 원하는 방향으로 힘을 작용하게 할 수 있음.

### GravityArea 동작 원리

- 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 동작 원리


- 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 오브젝트가 있을 수 있게 회전조절

### 중력의 퍼즐 요소

- RayCast 사용 기능
  - 중력 오브젝트를 토글
  - 퍼즐에 사용되는 'moveable' 태그 오브젝트의 초기 위치로 전환
- 특정 오브젝트와 충돌한 플레이어/오브젝트 저장된 위치/초기 위치로의 전환

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
		프로젝트 명	AI SENMA	
		팀 명	LAB 36	
		Confidential Restricted	Version 1.1	2024-05-24

### GravityController

- 기능: 마우스 포인터 위치에서 RayCast로 충돌 여부 확인.
- 'GravityObject' 태그 오브젝트와 충돌 시:
  - GravityHandler 스크립트의 ToggleGravity 함수 호출.
- 'moveable' 태그 오브젝트와 충돌 시:
  - 해당 오브젝트의 초기 위치와 회전 설정.

### GravityHandler

- ToggleGravity 함수:
  - RayCast로 Hit된 'GravityObject' 태그 오브젝트의 자식 중 'GravityArea' 존재 여부 확인.
  - 존재할 경우, GravityArea Scale 애니메이션 적용 Coroutine 수행.
    - 2초간 GravityArea의 Scale을 줄였다 커지게 하는 애니메이션 제공.
  - 중력에 영향을 받는 오브젝트는 애니메이션에 따라 다양한 방향으로 이동.
  - 게임의 변칙성을 제공하여 난이도 조절 가능.

### 중력 퍼즐 요소

- 중력 또는 다양한 힘의 영향을 받아 오브젝트를 원하는 공간에 도달시키기.
- 실패 시, 무중력 상태에서 오브젝트는 변칙적으로 계속 움직임.
- 여러 번 시도 가능하도록 초기 위치로 이동시키는 기능 추가.
  - 'moveable' 태그 오브젝트가 RayCast에 Hit되었을 때 실행.
  - 초기 위치로 되돌리며, Rotation을 0,0,0으로 초기화.
- Rigidbody의 isKinematic을 Off -> On으로 설정하여 물리적 작용 제거 후 초기 상태 복원.

### 플레이어 관련 요소


- SaveArea: 퍼즐 시작 지점마다 플레이어 위치 저장.
- Blackhole/BlackholeBall과 충돌 시: 저장된 위치로 이동.

### 오브젝트

- 'moveable' 오브젝트 RayCast 시 실행되는 기능과 동일.
- Blackhole/BlackholeBall과 충돌 시: 맵에 배치된 초기 위치로 이동.
- 공간 벗어나 RayCast 사용 불가 상황 방지.

### BackToSavedPos 스크립트

- 기능: 블랙홀과 충돌한 오브젝트가 플레이어 또는 오브젝트인지 구분하고 특정 위치로 전환.
- 목적: 중력을 이용한 퍼즐 요소 및 허용된 공간 내에서만 활동하도록 개발.
- 퍼즐의 구성요소로 함정, 장애물 등으로 활용.

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
		프로젝트 명	AI SENMA	
		팀 명	LAB 36	
		Confidential Restricted	Version 1.1	2024-05-24

## 프로젝트 진행 내용 - 암흑물질

### 암흑물질 발사 및 반사 구현

- raycasthit 함수를 이용하여 암흑물질이 물체와 충돌하거나 일정 거리만큼 나아간 위치를 검증함.
- 거울에 부딪히면 방향을 90도로 변경, 일반 물체에 충돌하면 암흑물질이 멈추도록 구성함.

### 암흑물질의 렌더링

- 암흑물질의 색깔, 굵기 등의 요소를 설정하여 렌더링 작업을 거침.
- 거울에 부딪히는 경우 각 위치들을 list에 담아서 기억하고 암흑물질이 멈출 경우 저장해둔 위치를 바탕으로 암흑물질을 렌더링함

### 색상 혼합 기능

- 포인터에서 나온 암흑물질의 색을 바탕으로 특정 벽에 닿은 경우 OnTriggerStay를 이용하여 부딪힘을 유지하고 있을 경우에만 색상이 업데이트 되도록 구현함.
- 암흑물질을 발사하는 기계를 이용하여 특정 벽에 맞추면 색상이 혼합되는 기능을 추가함.
- Wall Color 컴포넌트를 부착하고, 문을 지정하여 벽의 색깔과 일치하면 다음 스테이지가 열리게 설정함.

### 양면 거울 퍼즐

- 양면으로 반사되는 퍼즐 내용을 추가함.
- 프리팹의 단면 거울 반사체 부분을 복사하여 반대편에 이어 붙여 제작함.

 <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 1.1	2024-05-24

### 암흑물질 물리력 부여


- 암흑물질을 따라 box collider의 게임 오브젝트를 생성하게 제작하여 암흑물질이 충돌하도록 개발함.
- Laser Manager에서 모든 암흑물질을 관리하며, setActive로 불필요한 자원 소모를 줄임.

### 암흑물질 충돌 판정

- 여러 암흑물질이 동시에 발사될 때 충돌 판정을 관리하기 위해 암흑물질 매니저를 두고, 발사된 순서대로 유지되게 함.
- Laser Manager list에 담아두고 순차적으로 발사하게 하여 우선순위 문제를 해결함.

### 암흑물질과 플레이어 상호 작용

- 암흑물질에 실제 물리력을 부여하여 플레이어가 밟고 다닐 수 있게 개발함.
- box collider를 유지하여 플레이어가 실제로 걸어갈 수 있게 함.

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
		프로젝트 명	AI SENMA	
		팀 명	LAB 36	
		Confidential Restricted	Version 1.1	2024-05-24

## 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>	중간보고서		
		프로젝트 명	AI SENMA	
		팀 명	LAB 36	
		Confidential Restricted	Version 1.1	2024-05-24

### 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.


 <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 1.1	2024-05-24

### 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>	중간보고서		
		프로젝트 명	AI SENMA	
		팀 명	LAB 36	
		Confidential Restricted	Version 1.1	2024-05-24


## 프로젝트 진행 내용 - 스토리

### ChatManager.cs

- 기능: Unity 프로젝트에서 채팅 시스템 관리. TextMeshPro 사용하여 채팅 메시지 화면 표시 및 일정 시간 후 메시지 점진적 투명화.
- 구성요소 및 동작:
  - Awake() 메소드: 싱글톤 패턴 구현, DontDestroyOnLoad(gameObject) 사용.
  - Update() 메소드: 엔터키 누르면 MakeTextOpaque() 메서드 호출하여 메시지 불투명화.
  - AddMessage(string message) 메소드: 새로운 메시지를 messageQueue에 추가, ShowMessages() 코루틴 시작, 이전 메시지 흐리게 만드는 FadeOutText() 코루틴 다시 시작.
  - ShowMessages() 코루틴: messageQueue의 메시지를 하나씩 chatText에 추가, 최신 메시지를 표시하며 스크롤 위치 항상 맨 아래로 유지.
  - FadeOutText() 코루틴: 10초 대기 후 3초에 걸쳐 메시지를 투명하게 만듦.
  - MakeTextOpaque() 메서드: 모든 코루틴 중단, chatText의 모든 문자를 불투명화, FadeOutText() 코루틴 다시 시작.

### TutorialTrigger.cs

- 기능: 특정 트리거 영역에 플레이어가 들어올 때 메시지 표시. 트리거 발생 후 다시 발생하지 않음.
- 구성요소 및 동작:
  - Start() 메서드: triggerID 설정, PlayerPrefs 사용하여 트리거 발생 여부 확인.
  - OnTriggerEnter(Collider other) 메서드: 플레이어가 트리거 영역에 들어올 때 DisplayMessage 코루틴 시작.
  - DisplayMessage() 코루틴: messageDelay 후 ChatManager를 찾아 메시지 추가, hasTriggered를 true로 설정하고 PlayerPrefs에 저장.


	<b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서	
		프로젝트 명	AI SENMA
		팀 명	LAB 36
		Confidential Restricted	Version 1.1      2024-05-24

### FlashingTextTrigger.cs

- 기능: 특정 트리거 영역에 플레이어가 들어올 때 화면 중앙에 깜빡이는 텍스트 메시지 표시.
- 구성요소 및 동작:
  - Start() 메서드: 게임 시작 시 텍스트 숨김, triggerID 설정, PlayerPrefs 사용하여 트리거 발생 여부 확인.
  - OnTriggerEnter(Collider other) 메서드: 플레이어가 트리거 영역에 들어올 때 DisplayFlashingText 코루틴 시작.
  - DisplayFlashingText() 코루틴: 텍스트 활성화 후 displayDuration 동안 깜빡이게 함, 텍스트의 알파 값을 변경하여 페이드 인/아웃 효과 제공.
  - FadeTextToAlpha(float targetAlpha) 코루틴: 텍스트의 알파 값을 fadeDuration 동안 점진적으로 변경하여 페이드 인/아웃 효과 제공.

### DontDestroyOnLoadManager.cs

- 기능: 씬 전환 시 파괴되지 않는 게임 오브젝트에서 ChatManager 인스턴스를 찾아 반환.
- 구성요소 및 동작:
  - FindChatManagerInDontDestroyOnLoad 메서드: cachedChatManager 가 설정되어 있으면 반환, GetDontDestroyOnLoadObjects 메서드 호출하여 DontDestroyOnLoad 상태의 모든 루트 게임 오브젝트 가져옴.
  - GetDontDestroyOnLoadObjects 메서드: 새로운 임시 게임 오브젝트 생성 후 DontDestroyOnLoad 상태로 설정, 임시 오브젝트의 씬을 통해 DontDestroyOnLoad 상태의 씬을 가져와 루트 게임 오브젝트 배열로 반환, 임시 오브젝트 제거.

 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	AI SENMA	
	팀 명	LAB 36	
	Confidential Restricted	Version 1.1	2024-05-24

### 3 수정된 개발 내용 및 개발 과정

#### 3.1 문제해결 과정

##### 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 야기할 수 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

**문제:** 에셋에 발소리를 입히려는데 애니메이션상에 문제가 있어 발소리가 중복되는 문제가 생김

**해결방법:** 발소리를 애니메이션에 입히는 방법이 아니라 플레이어의 발이 땅에 닿을 때를 판단해서 소리가 나게 하는 방식으로 변경

**문제:** 플레이어의 점프가 버벅거리는 문제가 있었음

**해결방법:** 중력을 인위적으로 발생시키려고 한 문제로 특정 상황에만 중력을 설정하여 해결함

 <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 1.1	2024-05-24

## 염력 및 크기 조정

**문제:** 기능의 개발은 따로 어려운 사항은 없었고 실제 마우스의 위치와 움직인 물체의 위치가 멀어질수록 오차가 생기게 됨. 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

**문제:** 카메라가 기울어져있어 물체를 잡고 움직일 때 오차가 발생함을 알게 됨


**해결방법:** 물체를 드는 방식을 마우스드래그에서 플레이어고정으로 바꿈. 그에 따라 물체의 크기를 조절하면 필연적으로 물체가 가까워 지는 문제가 생기는데 이는 크기에 비례해서 물체의 거리를 조절하는 것으로 해결함

**문제:** 물체의 크기를 변경할 때 물체가 벽이나 바닥에 붙으면 물체가 벽을 통과해버리는 문제가 있음

**해결방법:** 물체의 거리를 조절하는 방식이 물체의 리지드바디를 옮기는게 아닌 트랜스폼 자체를 옮기는 방식에서 생긴 문제로 물체의 움직임을 리지드바디를 통해 조절함. 또한 물체의 크기가 점점 커질수록 플레이어와 비벼지는 현상이 나타나서 물체가 다른 사물과 충돌중일 때는 물체의 크기를 조절 불가능하게 설계함

**문제:** 물체를 잡고 움직일 때 물체의 움직임을 플레이어의 움직임을 따라오지 못하거나 힌트용으로 만들어둔 콜라이더에 막히는 현상이 있었음

**해결방법:** 물체의 이동속도를 조정하고 물체의 충돌조건 및 범위를 설정해서 해결함

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
		프로젝트 명	AI SENMA	
		팀 명	LAB 36	
		Confidential Restricted	Version 1.1	2024-05-24

## ○ 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

**문제:** 플레이어가 땅에 닿았는지 확인하는 Box의 높이 때문에 플레이어가 바닥 지형에 끼어 점프 후 즉시 멈추는 문제


**해결 방법:** 점프 시 끼어있는 y 방향 길이 이상을 강제로 이동시킨 후 점프 속도를 가하도록 수정

**문제:** NetworkRegisterList를 여러 작업자가 동시 수정하여 충돌이 발생하는 문제

**해결 방법:** 여러개의 NetworkRegisterList를 등록할 수 있게 수정하여 각 작업자가 자신의 NetworkRegisterList를 사용하도록 수정

**문제:** 중력장(Gravity Area)을 적용하면 다른 맵에서의 물체를 들거나 잡는데에 문제가 생김

**해결방법:** 중력장의 사용을 제한적으로 변경하고 다른 맵과 사용하는 프리팹들을 분리시킴

	<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 1.1	2024-05-24

**문제:** 블랙홀과 큐브 등 맵에서 수정했던 사항이 플레이 모드에선 적용되지 않음

**해결방법:** 맵에서가 아닌 프리팹 모드에서 수정사항을 적용함

### 이외 문제들

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

### 중력 기믹의 최소화

- 다른 기믹들과의 조화를 예상하고 중력 기믹과 맵을 개발했지만 최종 목표물인 모든 기믹을 합치는 것을 시도했을 때, 기능의 볼륨이 너무 크다는 것을 확인했음.
- 싱글 플레이로 완성해둔 중력 기믹을 모두 사용하는 것보다 다른 기믹들과의 조화를 위해 볼륨을 줄이고 퍼즐로서의 조화가 가능한 기믹을 개발했음.


### 중력 기믹의 최소화 - 해결 방안

플레이어의 무중력/중력 이동 기믹 및

플레이어의 중력 방향에 따른 이동 기믹 제거

- 오브젝트의 이동이 기본이 되는 퍼즐을 위한 기믹 추가 구현
- **BackToSavedPos**
  - 플레이어/오브젝트가 기존 위치로 이동
  - Blackhole, blackhole ball 등 다양한 장애물 구현하여
- **SubmitArea**
  - GravityArea를 갖는 퍼즐 클리어 오브젝트로 Area에 제출된 큐브가 3초 이상 머물러 있어야 퍼즐 클리어 가능
  - 오브젝트가 정확한 위치로 이동해야 클리어 가능하게 정확성 요구



 <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 1.1	2024-05-24

## 빛(암흑물질)

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

**해결방식:** 버그가 발생한 이유를 찾고 있으며 곧 수정될 예정임 -> 충돌되는 거울의 collider를 조절하는 방식으로 프리팹을 수정하여 해결하였음.

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

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

**문제:** 현재의 퍼즐 방식이 레이저와 맞지 않는다고 판단함.

**해결방식:** 논의 결과 기존에 빛의 컨셉에서 암흑물질의 컨셉으로 변경함.

## 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.

**문제:** 암흑물질을 발사하는 경우 매번 암흑물질들의 우선 순위가 달라서 부딪히는 판정이 복잡한 경우가 종종 발생함


**해결방식:** 암흑물질 매니저를 두어서 리스트로 각 암흑물질들의 순서를 관리하여 해결하였음.

**문제:** 물건을 차단하는 투명한 벽이 존재하는데 암흑물질이 통과하지 못하는 문제가 발생함.

**해결방식:** 레이어마스크를 설정하여 물건을 막는 벽의 레이어를 제외하고 암흑물질을 ray를 쏘도록 하여 해결함.

**문제:** 정답에 다가가는 힌트가 모자라다는 의견이 있었음

**해결방식:** 빛으로 벽의 색을 맞추는 곳 위에 표시하거나, 특정 포인트에 빛을 배치하여 그곳을 향해 접근하여 문제를 풀이할 수 있도록 유도하게 하였음.

	<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 1.1	2024-05-24

## 스토리

**문제:** 특정 지점을 지나면 일회성으로 메시지가 나와야 했으나, 메시지가 올바르게 표시되지 않음.

**해결방식:** 트리거 스크립트를 수정하여 메시지를 한 번만 표시하고, 트리거가 한 번 발생하면 다시 발생하지 않도록 설정.

**문제:** 텍스트가 추가될 때마다 스크롤뷰의 스크롤이 아래로 밀리지 않음.

**해결방식:** ChatManager 스크립트에서 메시지를 추가한 후 스크롤뷰의 verticalNormalizedPosition 을 0 으로 설정하여 스크롤이 항상 최하단에 오도록 함.

**문제:** 메시지가 한꺼번에 표시되어 자연스럽지 않음. 실제 대화처럼 보이게 하고 싶어서 방법을 찾아야함, 이탤릭체 같은 <br><b><i> 등도 같이 출력됨.


**해결방식:** ShowMessages 코루틴에 큐스택 messageQueue.Dequeue()를 활용하여 한 글자씩 출력하도록 설정. 태그는 조건문으로 태그가 없는 내용만 출력.

**문제:** 다른 씬으로 이동했다가 돌아오면 메시지가 다시 표시됨.

**해결방식:** PlayerPrefs 를 사용하여 메시지 상태를 저장하고 씬 이동 후에도 유지되도록 설정.

**문제:** 텍스트가 깜빡이는 시간을 다르게 설정할 수 없음.

**해결방식:** 깜빡이는 시간을 다르게 설정할 수 있도록 onTime, offTime, fadeDuration 변수를 추가하고, 페이드 인/아웃 시간을 설정.

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
		프로젝트 명	AI SENMA	
		팀 명	LAB 36	
		Confidential Restricted	Version 1.1	2024-05-24

## 네트워크 개발

**문제:** 게스트가 호스트가 가보지 못한 씬에 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

### 멀티 플레이를 고려하지 않은 싱글 플레이를 위한 기술 개발

- 중간 보고서 제출 때까지 멀티플레이에 대한 고려를 하지 않고 개발을 대부분 진행했기 때문에 network behaviour로의 전환 필요
- Netcode 라이브러리를 사용하여 네트워크로 전환
- Network Behaviour 클래스를 상속받아 네트워크 관련 행동을 정의

### 싱글에서 멀티 플레이로의 전환 - 해결 방안

- RPC를 사용해 서버와 클라이언트가 서로 특정 작업을 요청할 수 있게 하여 싱글로 로컬에서만 작동하던 GravityHandler, GravityController를 멀티플레이로써 사용할 수 있게 했음
- 이후 개발한 BackToSavedPos와 SubmitArea 등 두 플레이어 모두 네트워크로 영향을 받아야 하는 모든 기능을 구현 완료

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
		프로젝트 명	AI SENMA	
		팀 명	LAB 36	
		Confidential Restricted	Version 1.1	2024-05-24

## 4 중간보고 피드백 결과

### 4.1 중간보고 요약

#### 강점


- 현실적인 목표를 잘 설정하였다고 보입니다(개발 방향성).
- 네트워크 기반 멀티 유저 게임 진행 방식
- 게임의 기획에 대한 스토리보드나 아이디어보드가 먼저 구현된 이후 게임의 코드 개발과 병행되어야 할 것 같습니다.

#### 부족한 점

- "재미"라는 요소를 어떻게 개선 발전시킬지에 대한 전략이 명확하지 않습니다.
- 스토리를 기반으로 한 게임 기획이 아직 완료되지 않았음
- 스토리/아이디어 보드 등의 내용이 그림, 다이어그램 수준에서 보고서에 담기도록 준비해 주세요.

#### 조언

- 앱스토어 등록을 목표로 설정해볼 것을 권장합니다.
- 스토리를 기반으로 한 게임 기획을 완료하고 이를 기반으로 개발이 진행되어야 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 1.1	2024-05-24

## 4.2 중간보고 피드백 이행

### 재미라는 요소에 대한 정형화 문제

#### 해결 방안

재미라는 요소를 정형화하기 위한 방법으로 중간보고 당시에 많은 사람에게 피드백을 받는 방법이 제시되었습니다. 이 과정에서 베타 테스트에 대한 부분을 진행하기로 팀 내부에서 논의가 오갔습니다. 결과 팀 내부에서 “각자 3~5명 정도의 사람들에게 베타 테스트를 부탁하여 이에 대한 피드백을 받고 이 부분들을 수정하자!” 라고 결론지었습니다.

#### 베타 테스트 방식


**대상:** 각 팀원이 선정한 3~5인의 인원

**방식:** 최종 빌드파일을 공유하여 플레이 후 이에 대한 설문을 진행함

**플레이 타임 기준:** 최소 30분 이상

**설문 내역:** <https://docs.google.com/spreadsheets/d/1X1tXD2bp4fO3sW-skk0KUzvSO9pP7NdR5Ttu7ae8ESo/edit?usp=sharing>

**응답 인원 :** 17명

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
		프로젝트 명	AI SENMA	
		팀 명	LAB 36	
		Confidential Restricted	Version 1.1	2024-05-24

## 유저 피드백 정리 및 개선 사항

### 유저 평가

#### \*10점 만점 기준 평균점수\*

- 게임이 재미 있었나요? **7.1점**
- 게임을 진행하는데 설명이 충분 했나요? **5.4점**
- 게임 플레이 이후 판단하신 게임의 난이도를 알려주세요! **7.4점**
- 조작감은 어떠셨나요? **4.7점**
- 스토리는 어떠셨나요? **6.2점**
- 퍼즐의 구성은 어떠셨나요? **7.2점**
- AI Senma의 총평점수를 알려주세요! **6.5점**
- 가장 최근에 하신 게임의 총평점수를 알려주세요! **7.0점**  
EX) LOL, 로스트아크, 롤토체스, 메이플스토리, 엘든링 등
- 가장 최근에 하신 두번째 게임의 총평점수를 알려주세요! **7.1점**  
EX) LOL, 몬스터 헌터, 데스티니2, 타르코프, 포탈2, 문명6 등

### 분석

전체적인 평가는 최초로 테스트된 프로토타입임을 감안하였을 때 예상보다 높은 편에 속합니다. 전체적인 조작감과, 게임을 진행하는데 있어서 각각의 기믹들을 익히기에 충분하지 못한 설명을 한 것은 설문으로도 나타났습니다.

또한 수많은 온라인 게임들에 비하면 '재미' 측면에서 강점을 띄기는 힘들어 보입니다.

하지만 온라인 게임이 아닌 수많은 패키지 게임을 하는 사람들을 대상으로 조사한 결과는 "의외로 신선한 아이디어와 재미있는 기믹들이 있어 즐기기 좋은 게임이었다." 였습니다.

이러한 점에서 테스터 분들이 작성해주신 피드백들을 기반으로 게임을 개선시킨다면 충분히 더 재미있는 게임으로 발전한 가능성이 있다고 판단됩니다.

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
		프로젝트 명	AI SENMA	
		팀 명	LAB 36	
		Confidential Restricted	Version 1.1	2024-05-24

### 장점

- 생각하며 클리어하는 쾌감이 있음.
- 참신한 방법으로 퍼즐을 클리어하는 것이 재미있음.
- 퍼즐 푸는 재미와 흥미로운 퍼즐 요소.
- 타사 게임에 비해 짧지만 강렬한 인상을 남김.
- 다양한 퍼즐과 재밌는 기믹으로 신선함.
- 퍼즐 아이디어가 참신하고 다채로움.
- 창의적인 요소와 퍼즐의 적절한 배합.

### 단점

- 난이도가 어려워서 헤맬 때가 많음.
- 사운드가 빈약하고 조작감이 불편함.
- 조작감과 편의성이 부족하며 설명이 불충분함.
- 스테이지가 넓거나 좁아서 공간 배치가 불편함.
- 설명이 눈에 잘 들어오지 않음
- 플레이 민감도가 심함.
- 물건을 들고 던질 때 조작이 불편함.
- 게임의 목적성에 대한 설명이 부족함.

	<b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
		프로젝트 명	AI SENMA	
		팀 명	LAB 36	
		Confidential Restricted	Version 1.1	2024-05-24

### 발생한 버그 및 문제

- 캐릭터가 땅에 박히거나 화면이 갑자기 돌아가는 현상.
- 점프 기능이 제대로 수행되지 않음.
- 상자가 없어지거나 벽이 뚫리는 버그.
- 물건을 들고 움직일 때 투명벽을 뚫거나 천장을 걷는 버그.
- 암흑물질 관련 중력이 제대로 꺼지지 않음.
- 특정 구간에서 박스가 사라지거나 맵 밖으로 날아가는 버그.
- 인게임 설정 조작 불가능, 화면 비율 문제, 가시성이 낮은 설명.


### 개선 요청 사항

- 사운드 추가 및 유저 편의성 개선.-
- 조작감 개선과 스테이지 클리어 시 빠른 이동 개선.
- 시각적 가이드 추가와 대사 가독성 향상.
- 배경음악 및 능력 사용 시 사운드 추가.
- 에셋 배치와 UI 개선, 중요한 정보 더 잘 보이게 설명 추가.
- 특정 버튼으로 물건 던지는 조작 추가.
- 각종 버그 수정 및 리셋 기능 추가.
- 게임의 목적성과 설명을 더 명확히.

### 추가 피드백

- 초기 단계에서 사용자가 쉽게 이해할 수 있도록 시각적 가이드 추가 필요.
- 캐릭터의 움직임과 물건 조작의 편의성 개선.
- 게임의 배경과 스토리를 더 잘 전달하기 위한 시각적 및 사운드 효과 추가.
- 에셋 배치와 디자인적인 부분에서 개선 필요.
- 게임 설정과 조작 설명의 가시성을 높이고, 사용자 친화적인 인터페이스 제공.
- 스테이지별 리셋 기능 추가 및 진행 방식에 대한 설명 강화.



 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	AI SENMA	
	팀 명	LAB 36	
	Confidential Restricted	Version 1.1	2024-05-24

## 4.3 내부 피드백 사항

팀원이 7명이기에 발생한 문제

### 1. 형상관리 문제

저희 팀은 7명 모두 커밋을 하며 코딩을 하고 작업을 같이 진행하였습니다. 여기서 생긴 문제로 형상관리에 대해 모두가 안일하게 생각한 부분이 있었습니다. 여러명이 동시에 하나의 오브젝트를 건드리는 경우 A가 작업한 내용물을 B가 덮어씌우는 일이 빈번히 발생하였고 이 과정에서 작업을 복구하고 이를 다시 제자리로 돌리기까지의 문제가 발생하였습니다.

형상관리에 익숙한 인원이 형상관리에 익숙하지 않은 인원에게 형상관리를 하는 방법과 존재의 이유 등을 알려주고 서로에게 질문하며 성장한 결과 형상관리 문제는 발생하지 않게 되었습니다.

### 2. 소통의 부재


규모가 커질수록 체계를 잡고 각자의 분업과 각자의 작업들을 더욱 공유하고 이를 관리하는 인원이 필요했습니다. 하지만 각자의 생각이 다르고 각자의 개발 방향성이 다름에 따라 누구는 하드코딩으로 문제를 해결하고 누구는 객체지향을 사용하여 상속으로 문제를 해결하는 등 서로의 코드를 점점 더 이해하고 공유하기 힘들어졌습니다.

이러한 문제들을 해결하기 위해 주기적으로 코드리뷰와 서로의 코드를 설명하고 이 기능과 구현의 이유를 서로 체크하는 시간을 가지며 나아진 팀워크를 볼 수 있었습니다.

### 3. 개발을 합치는 단계에서의 애로사항

수많은 기능들이 개발되었고 이를 네트워크로 올리는 작업을 중간보고 이후로 진행하였습니다. 다만 여기에는 한가지 큰 문제가 있었는데 개발된 수많은 기능들을 하나의 씬으로 합치게 되는 경우 생각지 못한 버그들을 만나게 되었습니다.

모두가 베타테스팅을 진행하며 받은 피드백들이 실제로 저희가 만들었던 것들도 있으며 이 부분들이 개발내용을 합치는 과정에서 우선순위에 밀려 사용하지 못한 코드들이 정말 많음에 아쉬워했습니다. 실제 오브젝트 아웃라인, 벽면중력 이외에도 각 기믹별로 삭제된 부분이 정말 많습니다.

 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	AI SENMA	
	팀 명	LAB 36	
	Confidential Restricted	Version 1.1	2024-05-24

## 4.4 오픈 마켓 등록

### STEAM 등록 현황

#### STEAM 앱스토어 등록 과정 및 내용

- STEAM WORKS(스팀 개발자 전용 페이지) 가입
- 스팀웍스 파트너로 가입 후 게임 등록이 가능
- 게임 심의 및 상점 게시 등록
- 게임 등록비 지불(보증금 100달러)
- 판매자 및 상점 정보 등록
- 게임 빌드 체크리스트 등록 후 심의 승인
- 스토브 심의 대기중(현재)

#### STEAM 등록시 STOVE의 심의를 받아야 하는 이유

- 게임물을 유통시키거나 이용에 제공할 목적으로 게임물을 제작 또는 배급하고자 하는 사람은 당해 게임물을 제작, 배포 전에 게임위로부터 등급 분류를 받아야 함.
- 스팀에 게임을 등록 후 출시를 할 수 있으나 한국 지역을 제한 해야함.
- 게임위로부터의 등급 분류가 아닌 스토브에 자체 심의를 통해 허가를 받을 수 있음.
- 스토브의 심의 후, 스팀에 게임을 등록하고 스팀 내부 심사를 거쳐 등록이 완료됨.

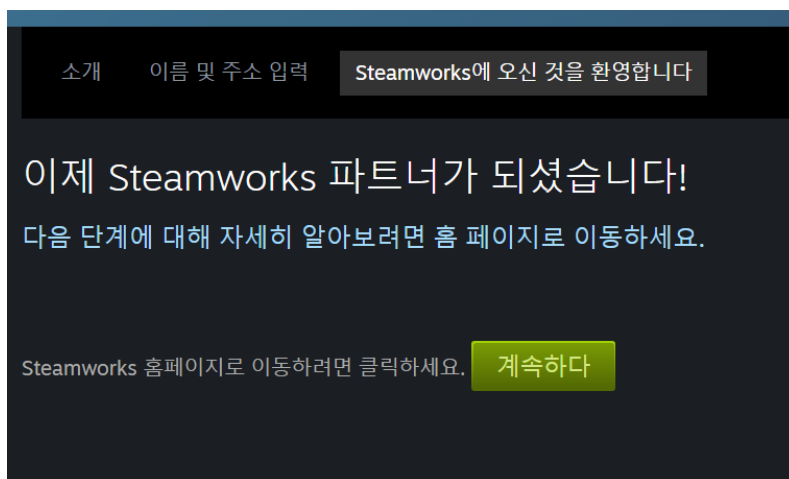


그림 3 Steamworks 파트너 등록 완료 사진

 <div>국민대학교 소프트웨어학부 캡스톤 디자인 I</div>	중간보고서		
	프로젝트 명	AI SENMA	
	팀 명	LAB 36	
	Confidential Restricted	Version 1.1	2024-05-24

Steamworks > App Admin

SENMA AI SENMA (3015680) Game

Release Progress:

Follow the instructions and checklists below to prepare for release. Your store page and game build may be worked on in any order. For more details, please see [Preparing for Your Release on Steam](#).

CHECKLIST COMPLETE

Your Store Presence

Your app is in the review queue. This review typically takes 2-3 days and we will e-mail you with feedback or approval. No further action is required until the review is complete.

You can view and respond to feedback on the Steamworks Support site [here](#).

Submitted for review on 23 May

CHECKLIST COMPLETE

Your Game Build

When you have a mostly final build of your game uploaded, select below to mark as ready for review.

Note: Your build will need to go through a quick review prior to posting. Please allow 3-5 business days.

Mark as ready for review...

Release

Once you've completed the following steps, your game will be ready for release:

- Store page and game build have been reviewed and approved
- Store page has been visible as 'Coming Soon' for at least two weeks
- Minimum of 21 days since the first purchase of an app credit

Your earliest possible release is two weeks after your store page has been visible as 'Coming Soon'. Please plan accordingly.

Store Packages, Pricing, & Release Dates [See Package Documentation](#)

These are the current, active, or future packages that might allow users to purchase or install this application.

PACKAGE TITLE	RELEASE DATE (?)	STATE (?)	PRICE (?)	ACTIVE DISCOUNT
AI SENMA - 1072558		Hidden	Free	

View Associated Items

All Associated Packages, DLC, Demos And Tools.

Manage store, product key and promotional packages. Add DLC, demos and tools.

Store Presence

Edit Store Page

Edit the description, add screenshots, trailers, art assets, etc.

View Store Page

View this application as it currently exists on the Steam store. ( [View in Beta mode](#) )

Specified date:

2024년 6월 7일

오전 12:59 GMT+9

Customers see:

Coming Soon

APP RELEASE STATE:

Unavailable (?)

LAUNCH DISCOUNT: (?)

None


그림 4 AI Senma 게임 등록 완료 및 심사 대기 사진

캡스톤 디자인 I

Page 43 of 45

중간보고서

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

 <div> 국민대학교 소프트웨어학부 캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	AI SENMA	
	팀 명	LAB 36	
	Confidential Restricted	Version 1.1	2024-05-24

## 스토브 스토어 등록 현황

### 스토브 스토어 등록 내용 및 방법

- <https://studio-docs.onstove.com/pc/GettingStarted/quickguide.html>

### 스토브 스토어 등록 과정

- 스튜디오 계정생성 및 등록 - 계약 - SDK연동/심의 - 상품 준비 및 출시
- 스토브스튜디오 등록 과정
  1. 스토브 스튜디오 가입 및 입점신청
  2. 프로젝트 등록 및 게임 상품정보 등록
  3. 게임 검수 및 심의

### 스토브 스토어 등록 현황

- 현재 스튜디오에 프로젝트 등록 완료 / 게임 등급분류 검수 중

### 스토브 스토어를 선택한 이유

- 스토브에서 지원되는 요소가 다양함.
- 상점페이지나 로고를 직접 디자인해주기도 하고, 영어관련 작업을 지원해주기도 하며, 마케팅과 등급심사도 지원해 줌.

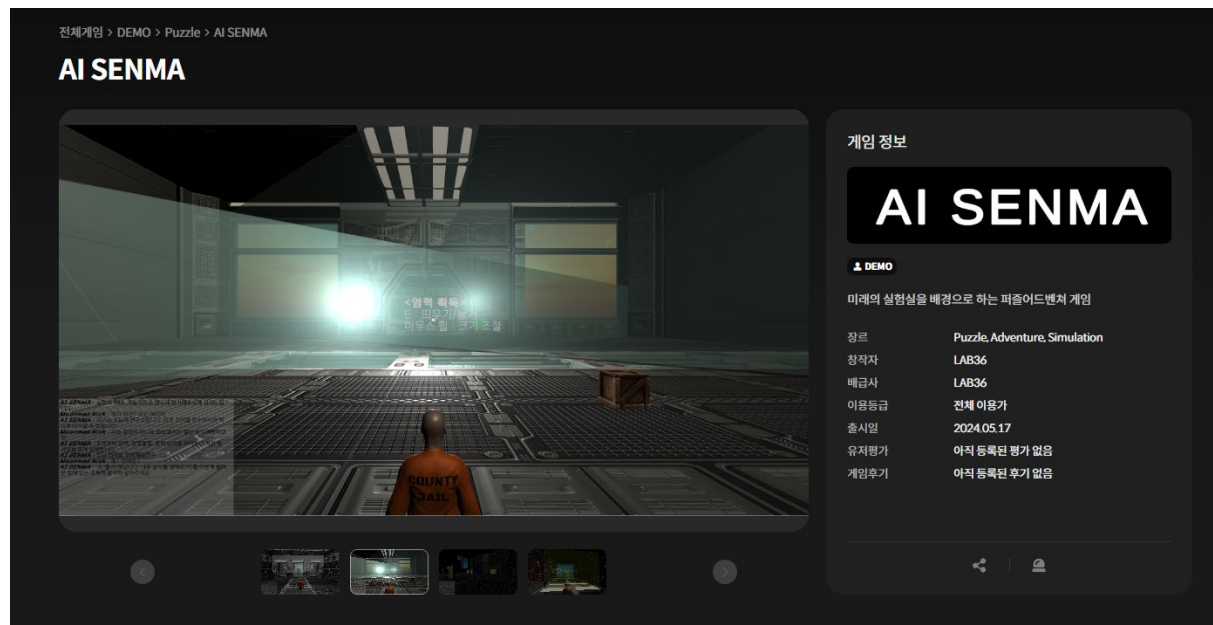


그림 5 Stove 스토어 게임 등록 사진

	<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 1.1	2024-05-24

## 5 향후 추진계획

### 5.1 향후 계획의 세부 내용

#### 스토어 등록 관련 향후 계획

- stove에서 심의결과가 나오는 대로 steam에 재등록하여 심사 이후 등록할 예정입니다.

#### 이외 향후 계획

- 피드백과 관련하여 많은 분들이 조금만 더 다듬고 발전시키면 참신한 아이디어와 기믹들의 조합으로 즐길만한 게임이 될 것 같다고 이야기 해주셨습니다. 이에 따라 기말발표 이후에도 패치를 이어 스팀에 실제로 등록하여 제대로 포트폴리오로 쓸 수 있도록 할 예정입니다.