

# 캡스톤디자인I 중간보고서

제 목	국문		객체 스토리지 서비스와 유니티 3D를 이용한 2인 멀티 퍼즐게임	
	영문		Two-person multi-puzzle game using object storage service and Unity 3D	
진행 상황	중요 마일스톤	기능 요구사항 1. 멀티 플레이어 기능 두 플레이어 간 상호작용이 가능하고, 실시간 서버 동기화가 일어나는 멀티 플레이어 기능 2. 다양한 구조와 서로 다른 로직의 퍼즐 맵 생성  3. 플레이어 조작 유저가 퍼즐 로직을 해결하기 위해 오브젝트를 이동하고 회전할 수 있는 조작 인터페이스  4. 플레이어 모션과 애니메이션 Unity 3D 모델링과 애니메이션을 사용하여 플레이어와 객체의 움직임을 구현  5. 플레이어간 소통 기능 게임 중 플레이어 간의 소통이 가능하도록 하는 기능  성능 요구사항 1. 맵 로드 최적화 2. 상호작용 응답시간 최적화 3. 자원 사용량 최적화 4. 동기화 시간 최소화  인터페이스 요구사항 1. 퍼즐 요소 / 게임 조작 방식 설명 가시성  데이터 요구사항 1.AWS를 이용한 데이터 관리		
		진행 상황 aws 서버를 활용하여 실시간으로 움직이는 오브젝트 구현 완료 기본적인 멀티플레이어 게임 로직 설계 완료 퍼즐 맵 디자인 작업 진행 중 퍼즐 요소 및 배경 테마 제작 중 플레이어 조작 - 퍼즐에 필요한 상호작용 요소 구성 키보드, 마우스 입력을 통한 퍼즐 상호작용 요소 및 인터페이스 구현 예정 플레이어 모션 - Asset Store를 활용한 요소 수집 캐릭터 모델링과 애니메이션 구현 중 UI 프로토타입 구성 완료		
산출물		요구사항 정의서(별첨 1), 중간보고서(별첨 2)		
팀 구성원	학년	학 번	이 름	연락처(전화번호/이메일)
	4	20201783	이정민	010-4595-7409 / jmdaniela0921@gmail.com

	3	20201759	김난희	010-6481-2507 / nh923915@gmail.c om
	4	20191790	최민준	010-9383-4133 / alswns567@naver.c om

컴퓨터공학과 의 프로젝트 관리규정에 따라 다음과 같이 요구사항 정의서와 중간보고서를 제출합니다

2024 년 5 월 2 일

책임자 : 이정민 (인)  
지도교수 : 이정 (인)

[별첨1]

프로젝트명 : 객체 스토리지 서비스와 유니티 3D를 이용한 2인 멀티 퍼즐게임

# 소프트웨어 요구사항 정의서

Version 1.0

개발 팀원 명(팀리더):

이정민(팀리더)

김난희

최민준

대표 연락처:010-4595-7409

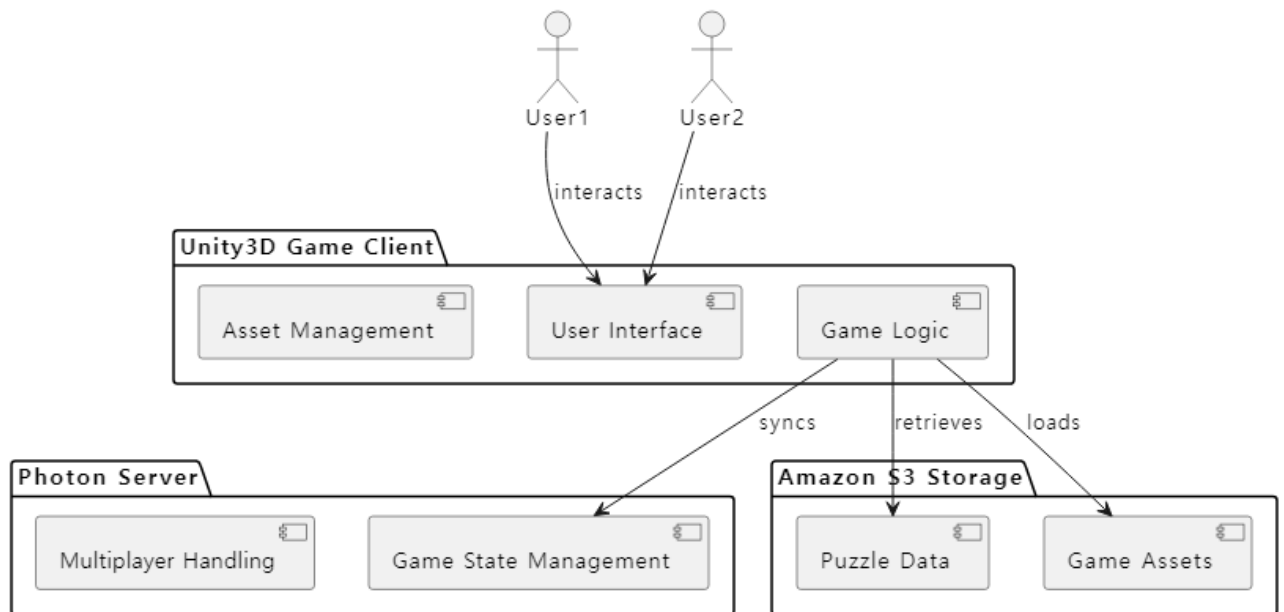
e-mail: jmdaniela0921@gmail.com

## 목차

1. 개요
2. 시스템 장비 구성요구사항
3. 기능 요구사항
4. 성능 요구사항
5. 인터페이스 요구사항
6. 데이터 요구사항
7. 테스트 요구사항
8. 보안 요구사항
9. 품질 요구사항
10. 제약 사항
11. 프로젝트 관리 요구사항

## 1. 시스템 개요

본 프로젝트는 사용자로 하여금 상호간 소통을 증진하고 문제 해결 능력을 키울 수 있는 2인 멀티 퍼즐 게임을 개발하는 것을 목표로 한다. Unity를 메인 엔진으로 사용하여 게임을 개발하고, Amazon S3 객체 스토리지에 데이터를 저장하고 자원을 관리한다. 또한 Photon 엔진을 통해 실시간 멀티플레이를 지원하여 안정적인 플레이를 보장한다.



## 2. 시스템 장비 구성요구사항

요구사항 고유번호		ECR - 001
요구사항 명칭		개발 엔진
요구사항 분류		시스템 장비 구성
요구사항 상세 설명	정의	게임 개발 엔진
	세부 내용	<ul style="list-style-type: none"> <li>- 장비 품목 : Unity</li> <li>- 장비 수량 : 라이선스 3개</li> <li>- 장비 기능 : 3D 모델링, 애니메이션, 물리 엔진, 크로스 플랫폼 개발 지원</li> <li>- 장비 성능 및 특징 : Unity Hub를 통한 프로젝트 관리 지원</li> </ul>

요구사항 고유번호		ECR - 002
요구사항 명칭		Amazon S3
요구사항 분류		시스템 장비 구성
요구사항 상세 설명	정의	클라우드 스토리지
	세부 내용	<ul style="list-style-type: none"> <li>- 장비 품목 : Amazon Simple Storage Service</li> <li>- 장비 수량 : 1</li> <li>- 장비 기능 : 스토리지 관리, 액세스 관리 및 보안, 데이터 처리</li> <li>- 장비 성능 및 특징 : 확장성, 데이터 가용성, 보안 및 성능 제공</li> </ul>

요구사항 고유번호		ECR - 003
요구사항 명칭		Photon
요구사항 분류		시스템 장비 구성
요구사항 상세 설명	정의	멀티플레이어 게임용 유니티 패키지
	세부 내용	<ul style="list-style-type: none"> <li>- 장비 품목 : Photon Unity Networking</li> <li>- 장비 수량 : 1</li> <li>- 장비 기능 : 호스팅, 매치메이킹</li> <li>- 장비 성능 및 특징 : 폴링, 캐싱 지원</li> </ul>

### 3. 기능 요구사항

요구사항 고유번호		SFR- 001
요구사항 명칭		퍼즐 스테이지 기능
요구사항 분류		기능
요구사항 상세 설명	정의	스테이지 기능 요구사항
	세부 내용	<ul style="list-style-type: none"> <li>- 1P : 첫 번째 플레이어의 역할은 색깔이 변화하는 공을 해당하는 색깔에 맞는 구멍에 넣음</li> <li>- 변화하는 색상의 공과 해당하는 여러 변수가 필요함</li> <li>- 2P : 두 번째 플레이어는 스테이지 구성 중 하나인 4줄짜리 책장에 꽂혀있는 책들을 확인하고, 그 사이에 색깔이 다른 책이 꽂혀있는 책이 몇 번째인지 확인</li> <li>- 퍼즐 스테이지 내부에 책장과 책들 등 퍼즐 관련 오브젝트 배치함</li> <li>- 게임 오브젝트의 Material 색상을 주기적으로 변경하는 기능</li> <li>- 충돌 이벤트가 발생할 시 색상에 따라 카운터를 증가시키는 기능</li> </ul>

요구사항 고유번호		SFR- 002
요구사항 명칭		퍼즐 스테이지 기능 2
요구사항 분류		기능
요구사항 상세 설명	정의	스테이지 기능 요구사항
	세부 내용	<ul style="list-style-type: none"> <li>- 1P : 3개의 큐브가 연결된 타워 2개에서 중복된 그림을 찾아 2P에게 전달해야 함</li> <li>- 2P : 1P에게 전달받은 그림이 그려진 오브젝트를 찾아 상호 작용해야함</li> <li>- 올바른 로직 상호작용시에만 퍼즐 단계를 통과할 수 있도록 기능을 구현함</li> </ul>

요구사항 고유번호		SFR- 003
요구사항 명칭		퍼즐 스테이지 기능 3
요구사항 분류		기능
요구사항 상세 설명	정의	스테이지 기능 요구사항
	세부 내용	<ul style="list-style-type: none"> <li>- 객체와 상호작용 시 객체의 상태가 변하는 기능</li> <li>- 각 플레이어들이 상호작용을 통해 오브젝트들을 대칭된 모양으로 배치하면 스테이지 클리어</li> </ul>

요구사항 고유번호		SFR- 004
요구사항 명칭		플레이어 간 상호작용 패턴
요구사항 분류		시스템 SW 요구사항 / 프로그램 요구사항
요구사항 상세 설명	정의	엔진 사용 관리
	세부 내용	<ul style="list-style-type: none"> <li>- Photon 엔진을 사용하여 플레이어 간 소통이 가능하게 하고, 이를 사용한 협응과정을 통해 퍼즐을 풀어나가게끔 구성해야 함</li> <li>- 키보드 입력에 따른 캐릭터 움직임</li> <li>- 마우스 입력을 받아, 선택된 객체가 특정 컴포넌트를 가지고 있으면 상호작용을 수행하는 기능</li> </ul>

요구사항 고유번호		SFR- 005
요구사항 명칭		TalkManager
요구사항 분류		기능
요구사항 상세 설명	정의	대화 데이터 관리
	세부 내용	<ul style="list-style-type: none"> <li>- 각 개체의 고유 ID에 따른 텍스트 배열 저장 기능</li> <li>- 초기 대화 데이터 생성 기능</li> <li>- 주어진 ID와 대화 인덱스에 따라 대화 텍스트 반환 기능</li> </ul>

요구사항 고유번호		SFR- 006
요구사항 명칭		GameManager
요구사항 분류		기능
요구사항 상세 설명	정의	전반 상호작용 관리
	세부 내용	<ul style="list-style-type: none"> <li>- 객체와의 상호작용을 시작하는 기능</li> <li>- 대화 인덱스에 따라 대화를 처리하고 UI를 표시하는 기능</li> <li>- 특정 상호작용을 종료하고 관련 객체를 비활성화하는 기능</li> </ul>



#### 4. 성능 요구사항

요구사항 고유번호		PER - 001		
요구사항 명칭		맵 로드 시간		
요구사항 분류		성능	응락수준	필수
요구사항 상세 설명	정의	목표 맵 로드시간		
	세부 내용	<ul style="list-style-type: none"> <li>- 플레이어가 처음 접속하거나 다음 맵으로 넘어갈 때, 서버의 로드시간을 의미함</li> <li>- 맵 로드시간: 사용자가 게임을 시작하거나 새로운 맵으로 이동할 때 소요되는 시간. 이는 사용자가 게임을 빠르게 시작하고 끊임 없이 플레이할 수 있도록 5초 이내로 유지되어야 함</li> <li>- 평균응답시간, 평균 처리시간, 평균 전송시간은 각각 맵 로드시간과 관련이 있을 수 있으며, 게임의 다른 기능에 대한 처리 속도와 시간 요구사항은 맵 로드시간에 영향을 줄 수 있음을 고려해야 함</li> </ul>		

요구사항 고유번호		PER - 002		
요구사항 명칭		상호작용 응답시간		
요구사항 분류		성능	응락수준	필수
요구사항 상세 설명	정의	목표 상호작용 응답시간		
	세부 내용	<ul style="list-style-type: none"> <li>- 플레이어가 목표 오브젝트와 상호작용하거나 어떠한 행동을 실행할 때, 이에 대한 응답시간을 의미함</li> <li>- 평균응답시간: 게임 서버로부터 클라이언트의 요청을 받아들이고, 해당 요청에 대한 응답을 반환하는데 걸리는 시간. 이는 100밀리초 이내로 유지되어야 함</li> <li>- 평균 처리시간: 클라이언트의 요청을 처리하고 게임 상태를 업데이트하는 데 걸리는 시간. 이는 게임의 복잡성과 크기에 따라 다를 수 있으나, 일반적으로 500밀리초 이내로 유지되어야 함</li> <li>- 평균 전송시간: 클라이언트와 서버 간의 데이터 전송에 소요되는 시간. 이는 네트워크 지연과 패킷 손실을 고려하여 50밀리초 이내로 유지되어야 함</li> </ul>		

요구사항 고유번호		PER - 003		
요구사항 명칭		자원 사용량		
요구사항 분류		성능	응락수준	필수
요구사항 상세 설명	정의	목표 자원 사용량		
	세부 내용	<ul style="list-style-type: none"> <li>- 메인 메모리 점유율: 게임 서버의 메인 메모리 사용량. 메모리 누수 및 메모리 부족으로 인한 서버 다운을 방지하기 위해 80% 이하로 유지되어야 함. 볼륨이 큰 게임이 아니므로 적정량의 최적 메모리 점유율을 유지해야 함</li> <li>- CPU 사용률: 게임 서버의 CPU 사용률. 게임 로직 및 네트워크 통신 처리를 위한 충분한 CPU 여유를 확보하기 위해 70% 이하로 유지되어야 함</li> <li>- 입출력 자원 사용률: AWS S3 서버와의 데이터 입출력 처리량. 대용량 데이터 전송 및 저장에 대한 효율적인 처리를 위해 60% 이하로 유지되어야 함</li> </ul>		

요구사항 고유번호		PER - 004		
요구사항 명칭		동기화 요소 관리		
요구사항 분류		성능	응락수준	필수
요구사항 상세 설명	정의	동기화 시간 관리		
	세부 내용	<ul style="list-style-type: none"> <li>- 상태 동기화 : 게임 서버는 두 플레이어 간의 게임 상태를 동기화하여 일관된 게임 경험을 제공해야 함. 각 플레이어의 위치, 속도, 상태, 게임 아이템 소지 여부 등을 실시간으로 동기화 해야 함</li> <li>- 입력 동기화 : 플레이어의 입력은 게임 서버를 통해 다른 플레이어에게 전달 되어야 함. 입력 동기화는 플레이어의 동작을 실시간으로 동기화하여 두 플레이어 간의 게임 상호작용을 보장</li> <li>- 시간 동기화 : 게임 서버는 두 플레이어 간의 게임 시간을 동기화하여 게임 이벤트가 동일한 시간에 발생하도록 보장해야 함.</li> <li>- 레이턴시 관리 : 플레이어 간의 네트워크 레이턴시를 관리하여 게임 서버가 플레이어의 입력을 즉시 반영할 수 있도록 해야 함. 레이턴시 관리는 게임 서버와 클라이언트 간의 통신 딜레이를 최소화하고, 플레이어 간의 경험을 향상 시키는 데 중요한 역할을 함</li> <li>- 위의 요소들을 통해 게임 플레이의 원활함과 일관성을 보장해야 함</li> </ul>		

## 5. 인터페이스 요구사항

요구사항 고유번호		SIR - 001		
요구사항 명칭		메인 메뉴		
요구사항 분류		인터페이스 요구사항	응락수준	필수
요구사항 상세 설명	정의	기본적인 게임 조작		
	세부 내용	<ul style="list-style-type: none"> <li>- 시작 버튼</li> <li>- 설정</li> <li>- 크레딧</li> <li>- 종료</li> </ul>		

요구사항 고유번호		SIR - 002		
요구사항 명칭		HUD(Heads - Up Display)		
요구사항 분류		인터페이스 요구사항	응락수준	필수
요구사항 상세 설명	정의	플레이 중 필요한 정보를 실시간으로 표현		
	세부 내용	<ul style="list-style-type: none"> <li>- 점수</li> <li>- 타이머</li> <li>- 인벤토리 아이콘</li> </ul>		

요구사항 고유번호		SIR - 003		
요구사항 명칭		대화 상자		
요구사항 분류		인터페이스 요구사항	응락수준	필수
요구사항 상세 설명	정의	정보, 대화 등을 제공		
	세부 내용	<ul style="list-style-type: none"> <li>- 텍스트 박스</li> <li>- 오브젝트 이미지</li> <li>- 선택지</li> </ul>		

요구사항 고유번호		SIR - 004		
요구사항 명칭		설정 및 옵션 메뉴		
요구사항 분류		인터페이스 요구사항	응락수준	필수
요구사항 상세 설명	정의	그래픽, 사운드, 컨트롤 설정 조정		
	세부 내용	<ul style="list-style-type: none"> <li>- 슬라이더</li> <li>- 체크박스</li> <li>- 드롭다운 메뉴</li> </ul>		

요구사항 고유번호		SIR - 005		
요구사항 명칭		게임 조작 방식 설명		
요구사항 분류		인터페이스 요구사항	응락수준	필수
요구사항 상세 설명	정의	기본적인 게임 조작		
	세부 내용	<ul style="list-style-type: none"> <li>- 처음 퍼즐 상호작용되는 요소와 상호작용 키에 대한 요소를 설명</li> <li>- 상호 작용 가능한 물체끼리 접촉될 시 올바른 상호작용인지를 사용자에게 전달</li> <li>- 플레이어에게 동기 부여 요소 등의 기초 사항 설명 필요</li> </ul>		

## 6. 데이터 요구사항

요구사항 고유번호		DAR - 001		
요구사항 명칭		데이터 관리		
요구사항 분류		데이터	응락수준	필수
요구사항 상세 설명	정의	데이터 관리 방안		
	세부 내용	<ul style="list-style-type: none"> <li>- AWS S3을 활용하여 데이터의 복사본을 저장하고 관리할 수 있음</li> <li>- 데이터 보존 요구에 따라 AWS S3의 스토리지 클래스를 선택하여 데이터를 저장하고 보존할 수 있음. 예로, 자주 액세스되지 않는 데이터의 경우 Glacier 스토리지 클래스를 사용하여 비용을 절약</li> <li>- AWS CloudWatch 및 AWS CloudTrail과 같은 모니터링 및 로깅 서비스를 사용하여 데이터의 품질을 관리하고 모니터링할 수 있음. 또한, AWS Glue와 같은 데이터 변환 및 준비 서비스를 사용하여 데이터의 품질을 유지하고 관리할 수 있음</li> <li>- 데이터 확장: AWS의 다양한 데이터베이스 및 스토리지 서비스를 활용하여 시스템이 확장 가능하도록 설계할 수 있음</li> </ul>		

요구사항 고유번호		DAR - 002		
요구사항 명칭		데이터 식별		
요구사항 분류		데이터	응락수준	필수
요구사항 상세 설명	정의	시스템에 사용될 내,외부 데이터 사용 정보 기술		
	세부 내용	<ul style="list-style-type: none"> <li>- 내부 데이터베이스: AWS RDS(Relational Database Service)를 활용하여 MySQL 또는 PostgreSQL과 같은 관계형 데이터베이스를 사용할 수 있음</li> <li>- 외부 데이터베이스: 주로 사용할 데이터베이스로는 AWS S3를 사용하여 객체 스토리지로 데이터를 저장할 수 있음. 또한, AWS DynamoDB를 사용하여 NoSQL 데이터베이스를 활용할 수 있음</li> </ul>		

## 7. 테스트 요구사항

요구사항 고유번호	TER - 001		
요구사항 명칭	테스트 유형		
요구사항 분류	테스트	응락수준	필수
요구사항 세부내용	<ul style="list-style-type: none"> <li>- 단위 테스트: 개별 컴포넌트 및 모듈의 기능 및 로직을 테스트하여 각각의 동작이 예상대로 작동하는지를 확인</li> <li>- 통합 테스트: 시스템 내 다양한 컴포넌트 및 모듈 간의 상호작용을 테스트하여 통합된 시스템이 예상대로 작동하는지를 확인</li> <li>- 시스템 테스트: 전체 시스템의 기능적 및 비기능적 요구사항을 테스트하여 시스템이 사용자의 요구를 충족하는지를 확인</li> <li>- 성능 테스트: 시스템의 성능을 측정하고 평가하여 목표한 처리량, 응답 시간 등의 성능 목표를 달성하는지를 확인</li> </ul>		

요구사항 고유번호	TER - 002		
요구사항 명칭	테스트 환경		
요구사항 분류	테스트	응락수준	필수
요구사항 세부내용	<ul style="list-style-type: none"> <li>- AWS 클라우드 환경을 사용하여 테스트 수행</li> <li>- 다양한 테스트 환경을 구성하여 사용자의 실제 환경과 유사하게 테스트</li> </ul>		

요구사항 고유번호	TER - 003		
요구사항 명칭	테스트 방법		
요구사항 분류	테스트	응락수준	필수
요구사항 세부내용	<ul style="list-style-type: none"> <li>- 자동화된 테스트 스크립트를 개발하여 테스트를 자동화하고 반복 가능한 테스트를 수행</li> <li>- 각 테스트 유형에 맞는 테스트 케이스를 작성하고 실행하여 시스템의 여러 측면을 테스트</li> <li>- 실제 사용자 시나리오를 모방하여 테스트를 수행하여 사용자의 실제 경험을 반영, 경험을 반영한 테스트를 통해 수정사항 분석</li> </ul>		

## 8. 보안 요구사항

요구사항 고유번호	SER-001
요구사항 명칭	보안 지침 준수
요구사항 분류	보안
요구사항 상세 설명	<ul style="list-style-type: none"><li>- 게임플레이에 필요 없는 기능의 권한이나 개인정보를 요청하지 않도록 함</li><li>- 게임 내에서 사용되는 데이터는 필요에 따라 암호화함</li></ul>

요구사항 고유번호	SER-002
요구사항 명칭	서버 보안 관련 규정 준수
요구사항 분류	DB 보안
요구사항 상세 설명	<ul style="list-style-type: none"><li>- VPC(Virtual Private Cloud)를 사용하여 게임에 사용될 네트워크를 격리하고 보호</li><li>- AWS WAF(Web Application Firewall)를 사용하여 웹 애플리케이션을 보호</li><li>- 데이터를 암호화하여 저장 및 전송. AWS KMS(Key Management Service)를 사용하여 키를 관리</li></ul>

## 9. 품질 요구사항

요구사항 고유번호		QR-001
요구사항 명칭		신뢰성
요구사항 분류		품질
요구사항 상세 설명	정의	품질관리
	세부 내용	<ul style="list-style-type: none"> <li>- 게임을 플레이하는 동안 진행이 막히거나 물체가 해당하는기능에 따라 작동하지 않는 현상이 없도록 해야함</li> <li>- 고장이 발생했을 때, 게임은 자동으로 복구되어야 함</li> </ul>

요구사항 고유번호		QR-002
요구사항 명칭		사용성
요구사항 분류		품질
요구사항 상세 설명	정의	품질관리
	세부 내용	<ul style="list-style-type: none"> <li>- 플레이어는 게임을 쉽게 배우고 플레이할 수 있어야 함</li> <li>- 게임은 직관적인 사용자 인터페이스를 제공해야 함</li> <li>- 음성 채팅 기능을 통해 플레이어들이 함께 퍼즐을 해결할 수 있어야 함</li> </ul>

요구사항 고유번호		QR-003
요구사항 명칭		유지관리성
요구사항 분류		품질
요구사항 상세 설명	정의	품질관리
	세부 내용	<ul style="list-style-type: none"> <li>- 게임에 새로운 퍼즐이나 콘텐츠를 추가할 때 변경처리 절차를 준수해야 함</li> <li>- 시스템에 발생한 버그나 문제는 신속하게 해결되어야 함</li> </ul>



## 10. 제약 사항

요구사항 고유번호		COR-001
요구사항 명칭		시스템 개발 제약사항
요구사항 분류		제약사항
요구사항 상세 설명	세부 내용	<ul style="list-style-type: none"><li>- 개발 엔진 : Unity</li><li>- 언어 : C#</li><li>- 개발 방법론 : Agile</li></ul>

요구사항 고유번호		COR-002
요구사항 명칭		설계 및 구현 제약사항
요구사항 분류		제약사항
요구사항 상세 설명	세부 내용	<ul style="list-style-type: none"><li>- 시스템 설계 및 구현에는 객체 스토리지 서비스를 사용해야 함</li><li>- 시스템은 클라우드 환경에서 구축되어야 하며, 해당 클라우드 서비스의 제약사항을 준수해야 함</li></ul>

요구사항 고유번호		COR-003
요구사항 명칭		asset 활용 제약사항
요구사항 분류		제약사항
요구사항 상세 설명	세부 내용	<ul style="list-style-type: none"><li>- 구매한 Asset의 라이선스 및 사용 규정을 엄격히 준수해야 함</li></ul>

## 11. 프로젝트 관리 요구사항

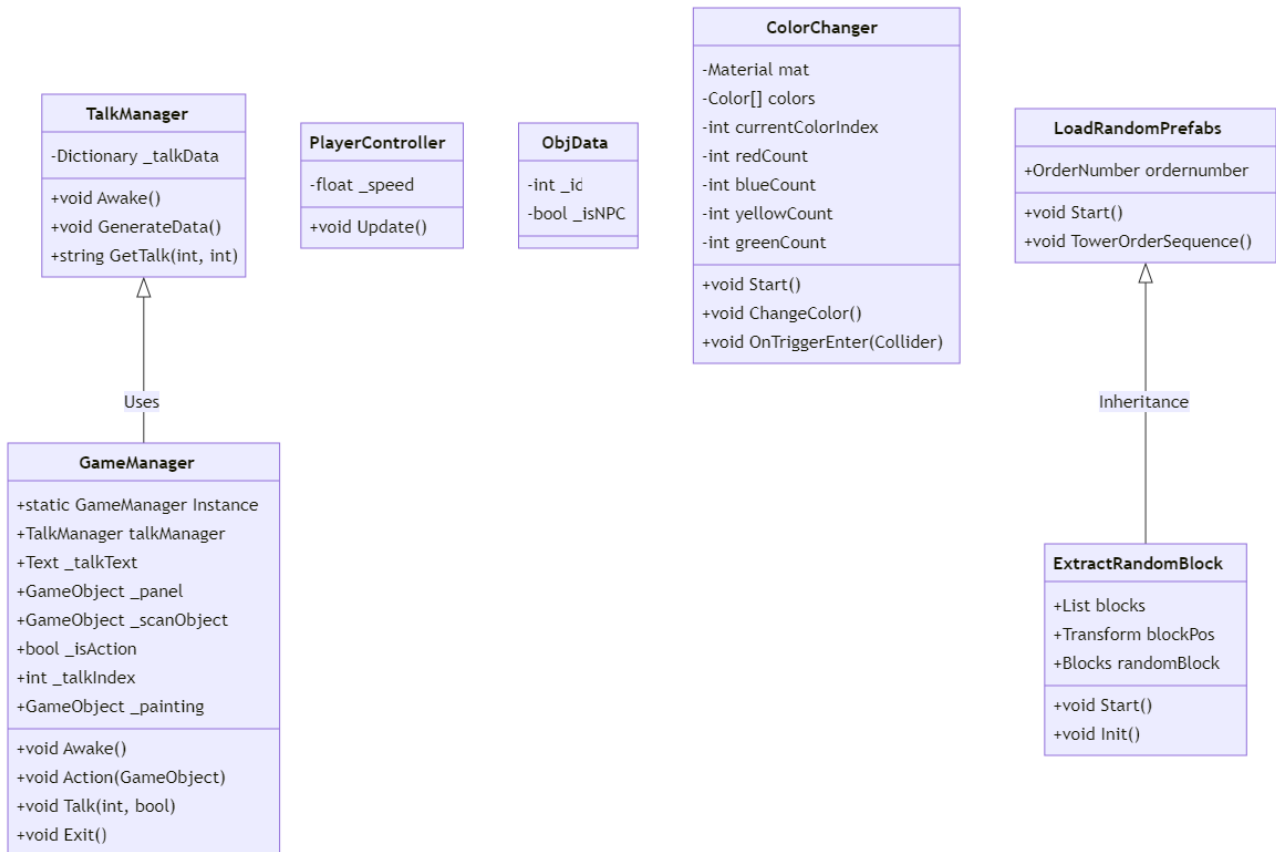
요구사항 고유번호		PMR - 001
요구사항 명칭		프로젝트 수행 조직
요구사항 분류		프로젝트 관리 요구사항
요구사항 상세 설명	세부 내용	<ul style="list-style-type: none"> <li>- 이정민 : UI개발, 스테이지 개발, Photon</li> <li>- 김난희 : 스테이지 개발, Amazon S3</li> <li>- 최민준 : 스테이지 개발, Photon</li> </ul>

요구사항 고유번호		PMR - 002
요구사항 명칭		프로젝트 일정 계획
요구사항 분류		프로젝트 관리 요구사항
요구사항 상세 설명	세부 내용	<ul style="list-style-type: none"> <li>- UI개발 : 5월 2주차까지 완료</li> <li>- 스테이지 개발 : 5월 말까지 완료</li> <li>- 멀티플레이 구현 : 6월 초까지 완료</li> </ul>

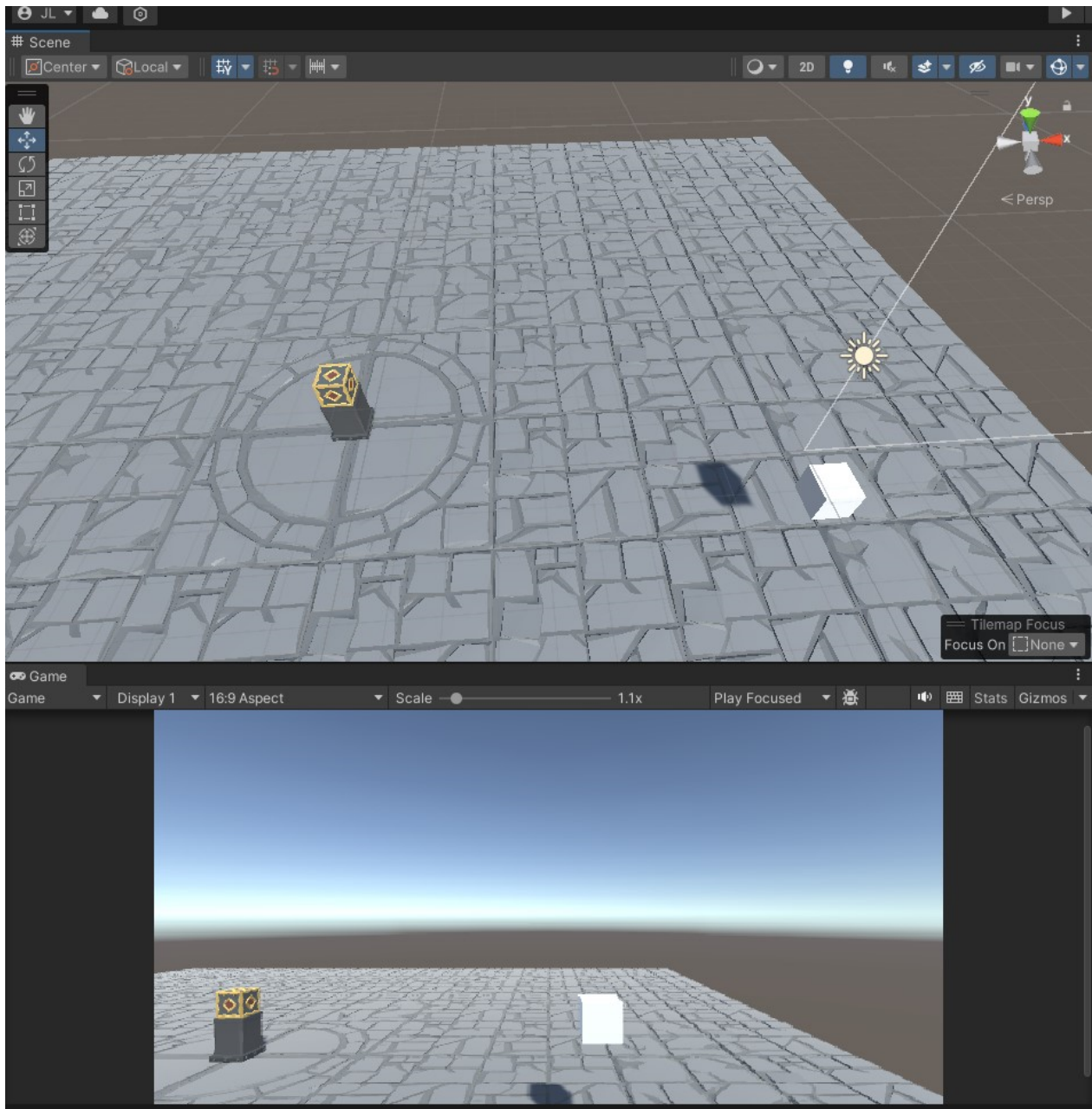
[별첨2]

## 중간보고서

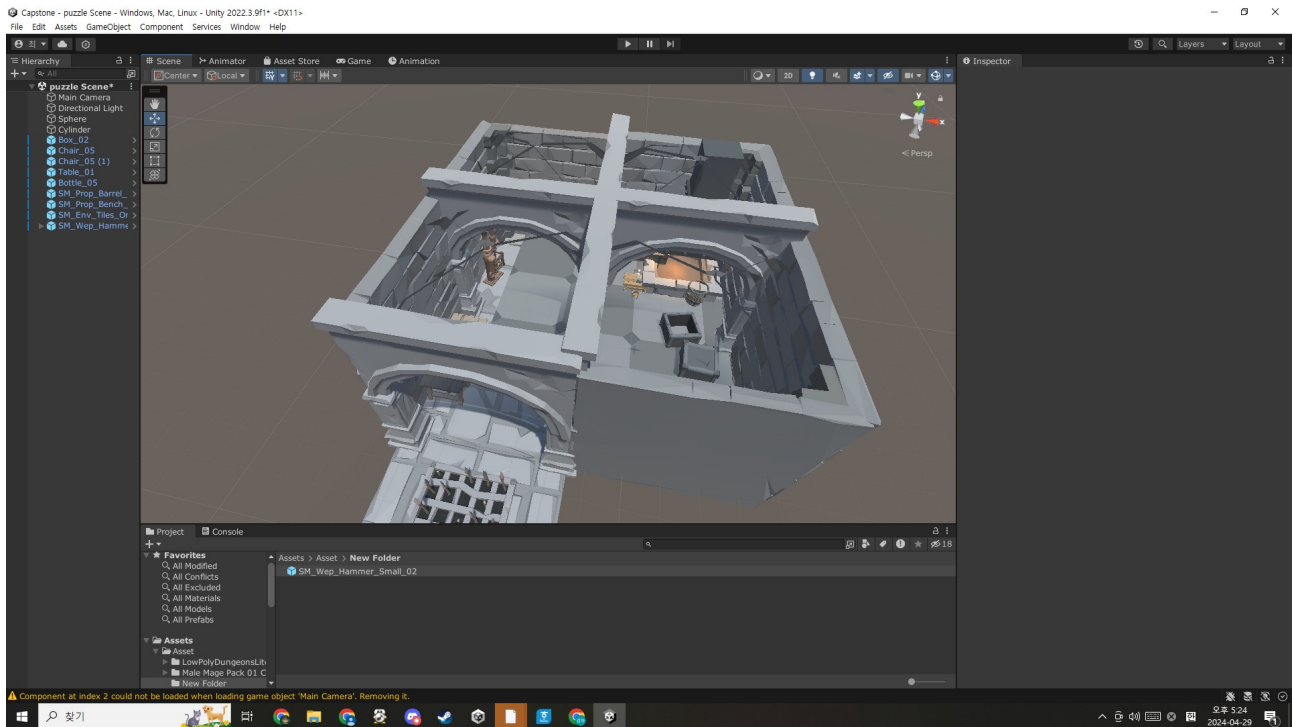
1. 요구사항 정의서에 명시된 기능에 대하여 현재까지 분석, 설계, 구현(소스코드 작성) 및 테스트한 내용을 기술하시오.



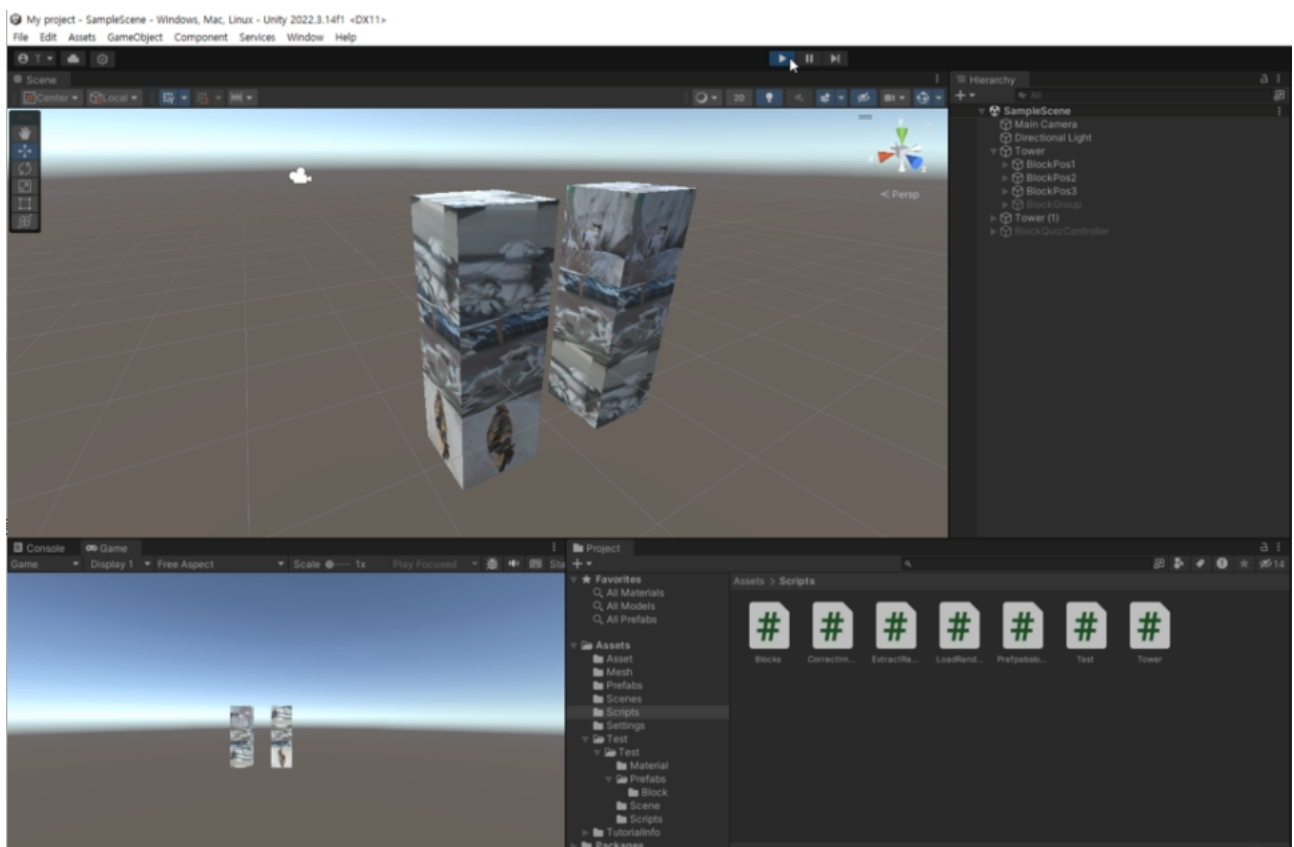
분석, 설계, 구현 및 테스트 클래스 다이어그램



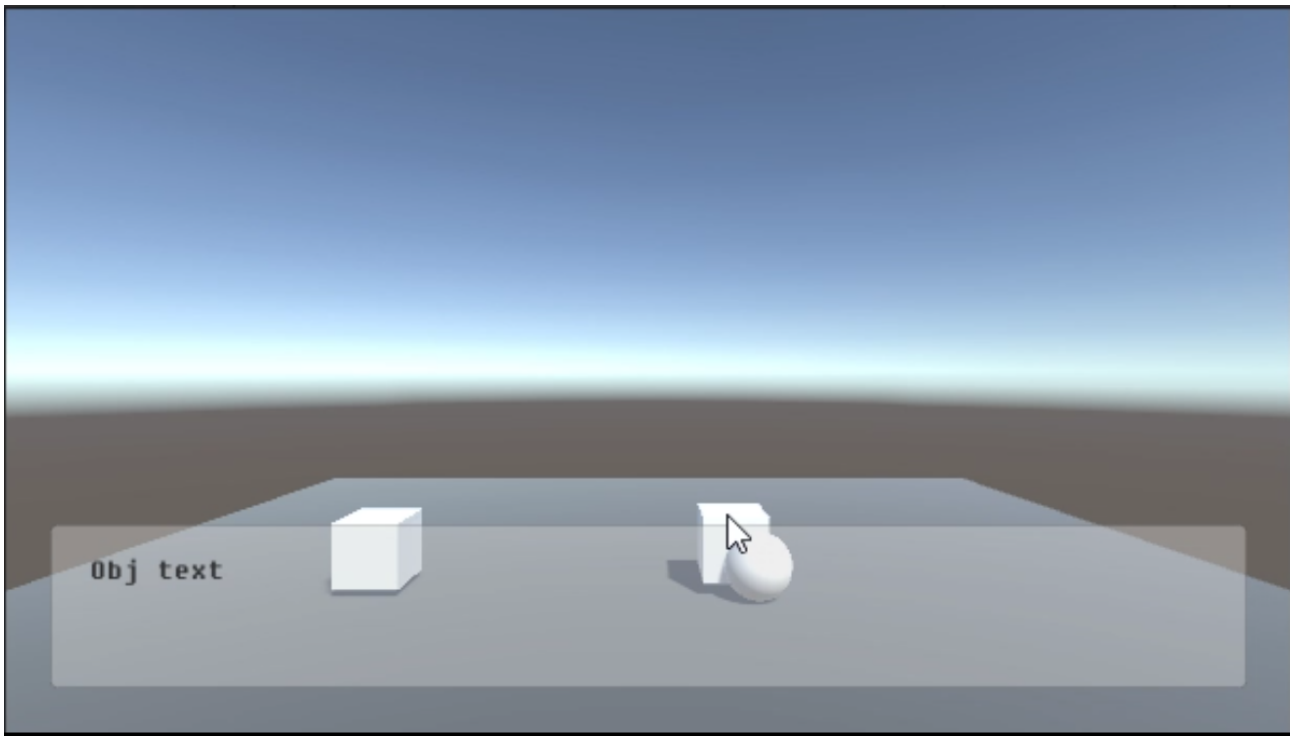
Asset 적용 예시 1



Asset 적용 예시 2



스테이지 기능 구현 예시



UI 구현 예시

```

5  public class ColorChanger: MonoBehaviour
13  [SerializeField] private int blueCount = 0;
    참조 2개
14  [SerializeField] private int yellowCount = 0;
    참조 2개
15  [SerializeField] private int greenCount = 0;
16
    참조 0개
17  void Start()
18  {
19      mat = GetComponentInChildren<MeshRenderer>().material;
20      InvokeRepeating("ChangeColor", 0f, 1f);
21  }
22
    참조 0개
23  void ChangeColor()
24  {
25      currentColorIndex = (currentColorIndex + 1) % colors.Length;
26      mat.color = colors[currentColorIndex];
27  }
28
    참조 0개
29  private void OnTriggerEnter(Collider other)
30  {
31      Color selfColor = mat.color;
32
33      if (selfColor == Color.red)
34          redCount++;
35      else if (selfColor == Color.blue)
36          blueCount++;
37      else if (selfColor == Color.yellow)
38          yellowCount++;
39      else if (selfColor == Color.green)
40          greenCount++;
41  }

```

```

8  public class GameManager : MonoBehaviour
25
    참조 0개
26  public void Action(GameObject scanObj)
27  {
28
29      _isAction = true;
30      _scanObject = scanObj;
31      ObjData objData = _scanObject.GetComponent<ObjData>();
32
33      Talk(objData._id, objData._isNPC);
34      _panel.SetActive(_isAction);
35  }
    참조 1개
36  void Talk(int id, bool isNPC)
37  {
38      string talkData = talkManager.GetTalk(id, _talkIndex);
39
40      if (talkData == null)
41      {
42          _isAction = false;
43          _talkIndex = 0; // 대화가 끝날 때 0으로 초기화
44          _panel.SetActive(_isAction);
45          return;
46      }
47
48      if (isNPC)
49      {
50          _talkText.text = talkData;
51      } else
52      {
53          _talkText.text = talkData;
54      }
55
56      _isAction = true;
57      _talkIndex++;
58  }
59
    참조 0개
60  public void Exit()
61  {
62      _painting.SetActive(false);
63  }
64  }
65

```

```

System.Random random = new System.Random();
int blocksLength = blocks.Count;
int randomInt = random.Next(0, blocksLength);

if (orderNumber == OrderNumber.A)
{
    //Tower1일때 3개 지정
    //랜덤 블록 지정
    randomBlock = Instantiate(blocks[randomInt]);

    //위치 지정
    randomBlock.transform.SetParent(blockPos);
    randomBlock.transform.localPosition = Vector3.zero;
    randomBlock.gameObject.SetActive(true);
}
else
{
    List<int> selectedIndices = new List<int>();

    // 3가지 블록을 랜덤하게 선택
    for (int i = 0; i < 3; i++)
    {
        int newIndex;
        // 중복되지 않는 블록 인덱스 선택
        do
        {
            newIndex = random.Next(0, blocksLength);
        } while (selectedIndices.Contains(newIndex)); // 이미 선택된 블록이라면 다시 선택

        selectedIndices.Add(newIndex); // 선택된 블록 인덱스 저장

        // 선택된 블록을 인스턴스화하여 위치 지정
        randomBlock = Instantiate(blocks[newIndex]);
        randomBlock.transform.SetParent(blockPos);
        randomBlock.transform.localPosition = Vector3.zero;
        randomBlock.gameObject.SetActive(true);
        //중복 제거하고 Tower1의 세가지 중 하나 따오기
    }
}

```

현재 플레이어 이동 및 상호작용 구현 완료하였으며 각 스테이지별 기능 구현과 Asset 적용 중에 있음. 또한 Photon을 적용하여 멀티플레이 구현 중에 있음.

2. 프로젝트 수행을 위해 적용된 추진전략, 수행 방법의 결과를 작성하고, 만일 적용과정에서 문제점이 도출되었다면 그 문제를 분석하고 해결방안을 기술하시오.

- 1) 애자일 방법론을 채택하려 했으나, 주기적인 스프린트 계획과 기능 배정에 대한 합의가 부족하여 애자일 방법론을 제대로 적용하지 못했음.
  - 일정 관리와 스프린트 계획을 강화하고, 주간 회의를 통해 각 팀원의 진행 상황을 점검하도록 함.
- 2) 계획된 개발 일정이 밀려 프로젝트 전반의 마일스톤 달성이 지연됨
  - 주간 회의 이외에 중간 평가를 통해 현재 진행 상황과 남은 과업을 점검하고, 일정을 조정하여 핵심 기능에 우선순위를 두고 개발하려 함.