

씨리얼 7조

TEXT 게임

RPG



GAME START!

목차

1. 팀 소개

2. 클래스 구조

3. 게임 시연 영상

4. 구현 세부사항

5. 테스트 케이스

6. 기능 체크리스트

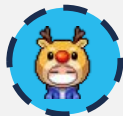
★ 왜 씨리얼인가요?

C++

+

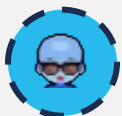
언리얼

팀원 소개 및 담당



캐릭터 및 스텟
스킬 시스템

팀장 지수빈



UI 구현
(아스키 아트)

부팀장 이찬호

```
  /\_/\
 (  o.o  )
  > ^ <
```



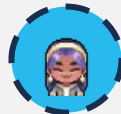
몬스터

팀원 정관빈



아이템

팀원 이근형

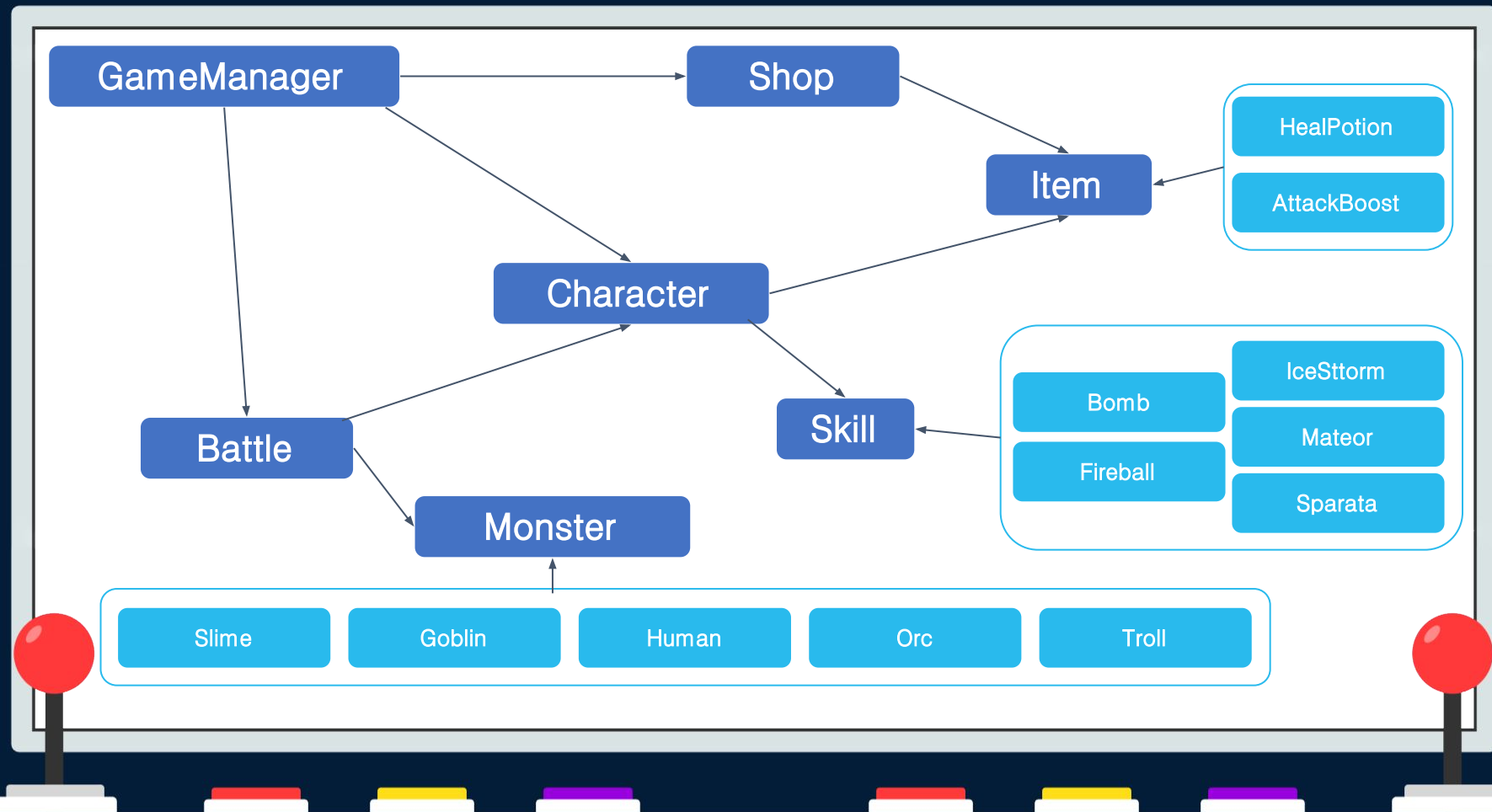


상점

팀원 남선우

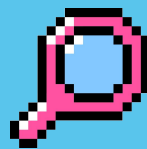


클래스 구조





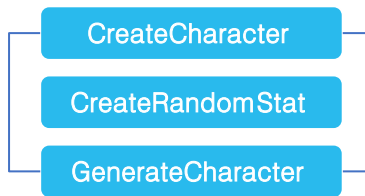
캐릭터 및 전투



캐릭터 생성 및 스탯 설정



Character Manager

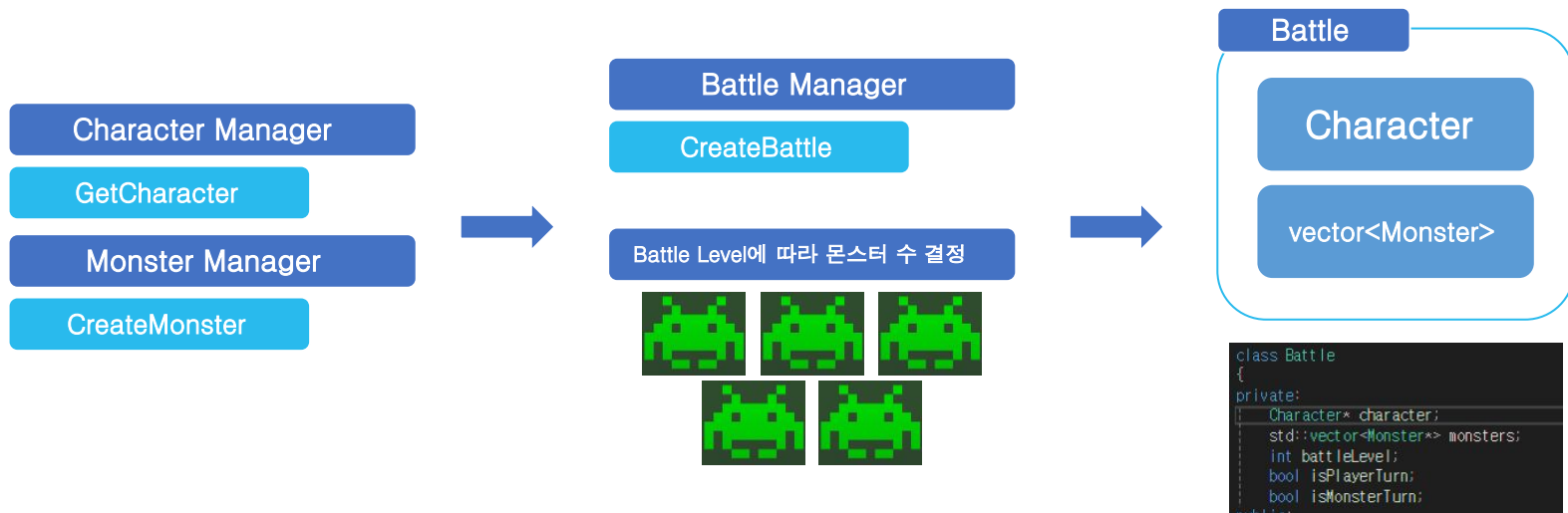


```
void CreateCharacter();           // 캐릭터 생성  
Character* GetCharacter();       // 현재 캐릭터 가져오기  
void CreateRandomStat(Character& character); // 캐릭터 스탯  
void GenerateCharacter(Character& character, bool isFirst); // 스탯 기반 캐릭터 설정
```

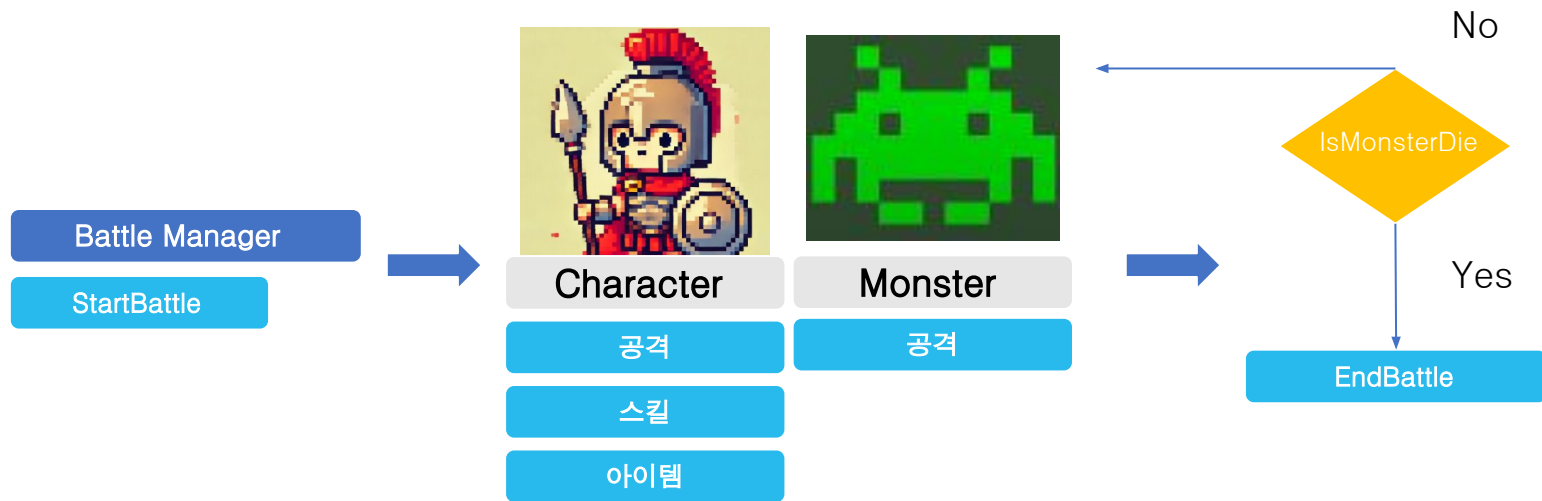
캐릭터 스킬



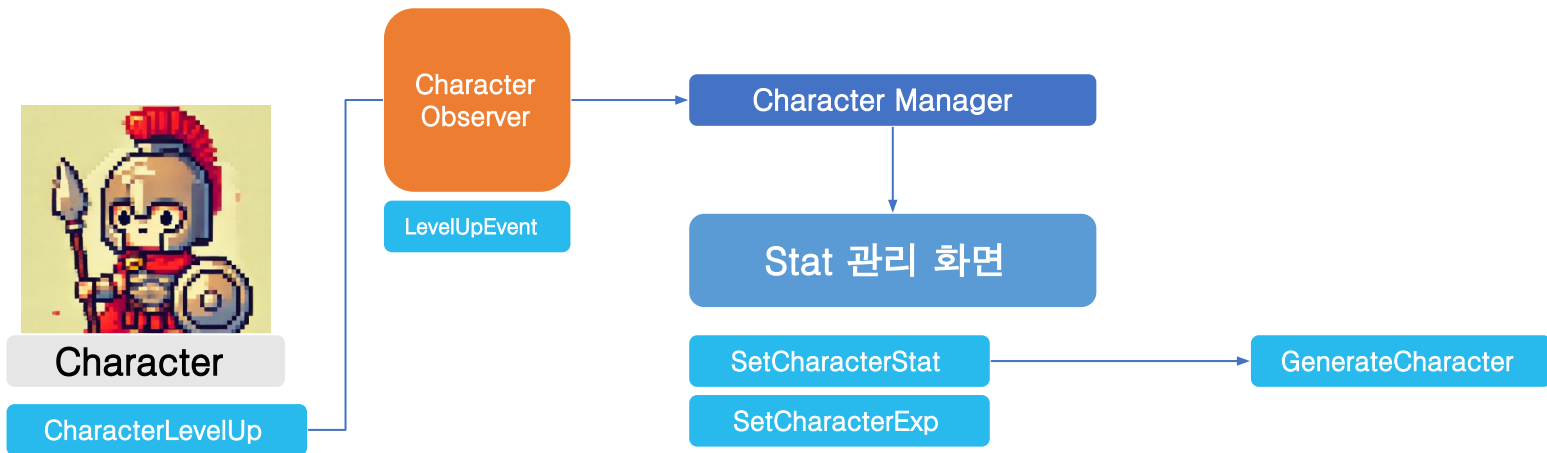
게임 전투 생성



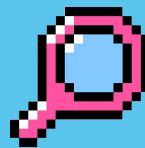
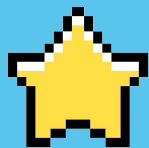
게임 전투 진행



캐릭터 레벨업



UI 및 로그 출력



메인 화면



1. 전투



2. 스탯



3. 인벤



4. 상점



5. 종료

| 입력 : |

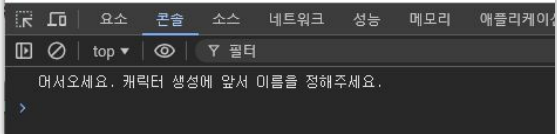
입력 칸

ㅇㅅㅇ 님 어서오세요. 원하시는 행동을 선택해주세요!
1. 전투 / 2. 스탯관리 / 3. 인벤토리 / 4. 상점 / 5. 종료

로그 출력

Text RPG

입력 :



TEXT RPG

입력 :

[여시오세요. 캐릭터 생성에 앞서 이름을 정해주세요.]



HTML을 참고해서
UI 제작

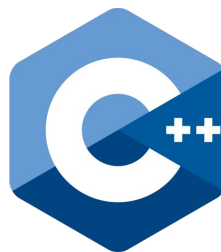


```
<!DOCTYPE html>
<html>
<body>
  <h1>Text RPG</h1>
  <input type="text" id="name" placeholder="입력 :">
</body>
<script>
  console.log("어서오세요. 캐릭터 생성에 앞서 이름을");
</script>
</html>
```

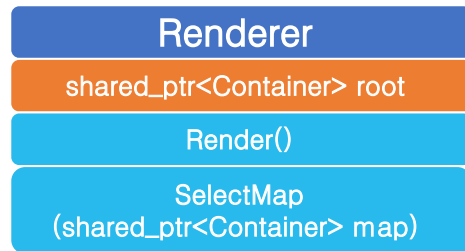
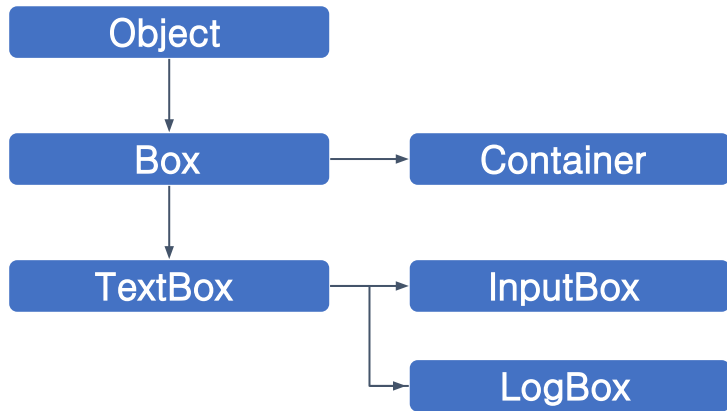
```
shared_ptr<Container> GameMap::Index()
{
    shared_ptr<Container> root = make_shared<Container>();

    shared_ptr<Object> title = make_shared<Object>();
    // 아스키 글자 소스 https://patorjk.com/software/taag/
    title->LoadFromTxt("title");
    title->SetColor(0, 255, 0);
    title->SetPos(12, 5);
    root->AddObject(title);

    return root;
}
```



Renderer 클래스가 렌더링할 수 있는 객체



Renderer

shared_ptr<Container> root

Render()

SelectMap
(shared_ptr<Container> map)

LogBox

Print
(string text,
RGB color)

RGB

uint8 r

uint8 g

uint8 b

unsigned __int8

InputBox

Input()

InputNumber()

렌더링 할 페이지

```
namespace GameMap
{
    shared_ptr<Container> Index();
    shared_ptr<Container> SelectStat();
    shared_ptr<Container> Main();
    shared_ptr<Container> SelectBattle();
    shared_ptr<Container> Battle();
};
```

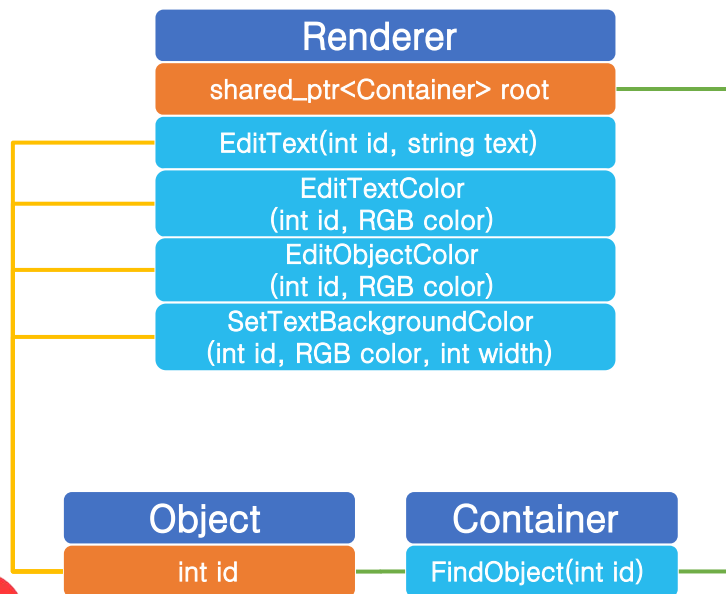
```
bool GameManager::Main()
{
    renderer->SelectMap(GameMap::Main());
    LogBox::GetInstance()->Print(format("{} 님 어서오세요. 원하시는 형

    int choice = 0;

    LogBox::GetInstance()->Print("1. 전투 / 2. 스텟관리 / 3. 인벤토리

    choice = InputBox::GetInstance()->InputNumber();
```

오브젝트에 데이터 주입



//캐릭터 체력바 렌더링

```
int hp = Character::GetInstance()->GetHp();
int mp = Character::GetInstance()->GetMp();
int exp = Character::GetInstance()->GetExperience();
int maxHp = Character::GetInstance()->GetMaxHp();
int maxMp = Character::GetInstance()->GetMaxMp();
int maxExp = Character::GetInstance()->GetExperienceCapacity();
int currentHp = hp * 80 / maxHp;
int currentMp = mp * 80 / maxMp;
int currentExp = exp * 80 / maxExp;

Renderer::GetInstance()->EditText(21, format("{} / {}", hp, max
Renderer::GetInstance()->EditText(22, format("{} / {}", mp, max
Renderer::GetInstance()->EditText(23, format("{} / {}", exp, ma

Renderer::GetInstance()->SetTextBackgroundColor(21, RGB(192, 0,
Renderer::GetInstance()->SetTextBackgroundColor(22, RGB(0, 0, 1
Renderer::GetInstance()->SetTextBackgroundColor(23, RGB(0, 192,
```

입력 : |

오소은 50 골드를 획득하였습니다!

오크 ★ 토벌완료 ★

트롤 | 35 HP | 7 AD

오크 | 38 HP | 4 AD

트롤 | 38 HP | 8 AD

슬라임 | 7 HP | 1 AD

★ 플레이어 턴 ★

1. 공격 / 2. 스킬 / 3. 아이템 / 4. 도망

입력 : |

오소은님이 3 만큼 데미지를 입었습니다!

오크 ★ 토벌완료 ★

트롤 | 3 HP | 7 AD

오크 | 38 HP | 4 AD

트롤 | 38 HP | 8 AD

슬라임 | 7 HP | 1 AD

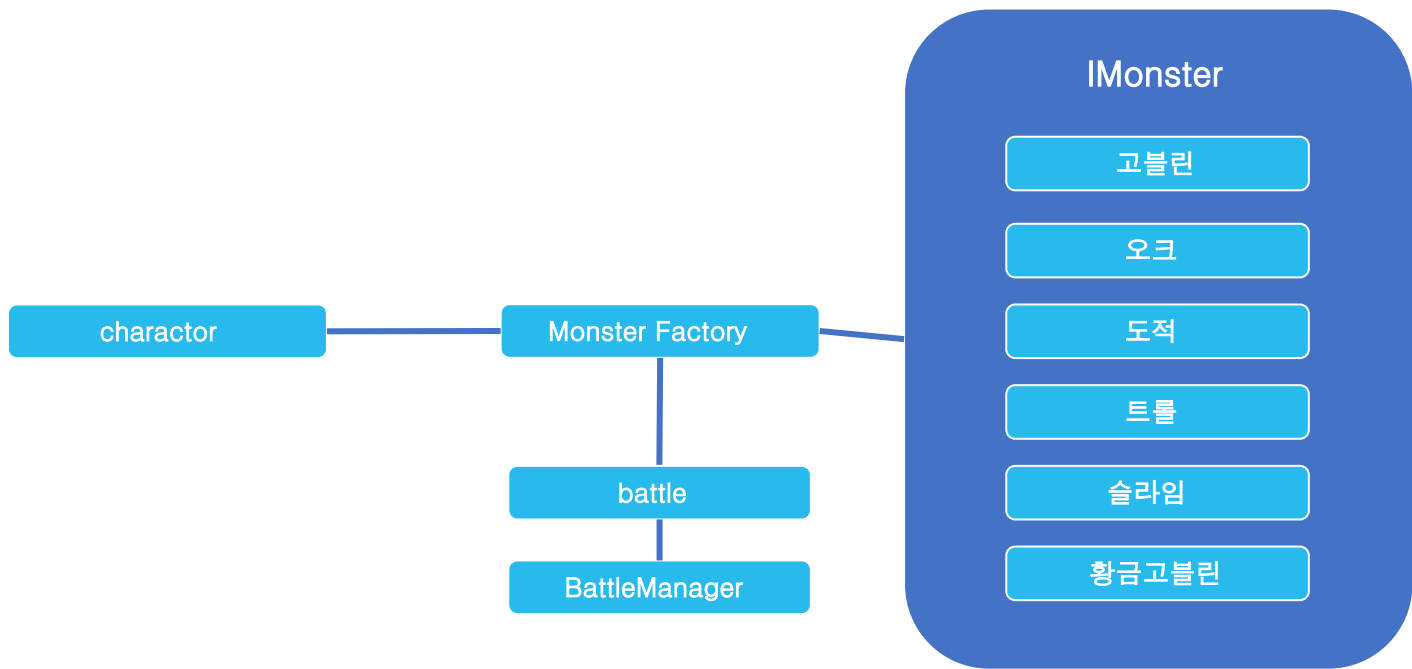
★ 플레이어 턴 ★

1. 공격 / 2. 스킬 / 3. 아이템 / 4. 도망



몬스터

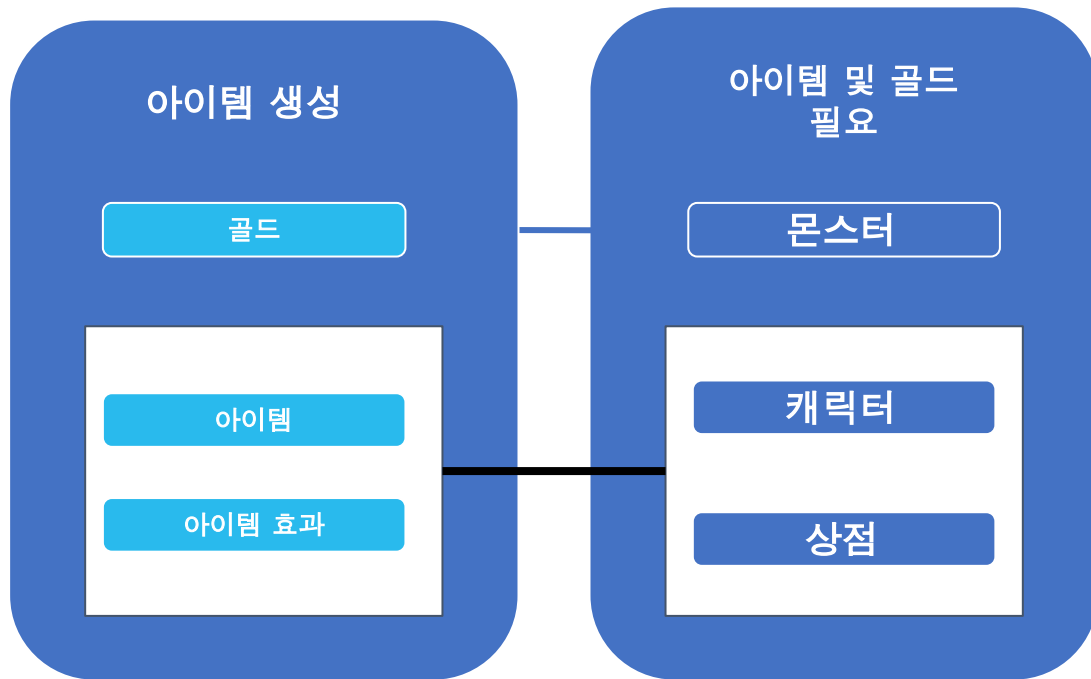




★ 아이템 매니저 🔍



아이템 매니저의 역할





상점



상점 방문

캐릭터

골드

인벤토리



상점

상점 인터페이스

구매 / 판매

재고 관리

아이템

아이템 정보

가격 / 효과



상점 방문 출력화면

*** 상점 방문을 환영합니다! ***

===== 상점 아이템 목록 =====

1. 체력 포션 (체력 +50, 공격력 +0) - 구매가: 10 골드, 판매가: 6 골드
2. 공격력 버프 포션 (체력 +0, 공격력 +10) - 구매가: 15 골드, 판매가: 9 골드

무엇을 하시겠습니까? (B: 구매, S: 판매, I: 인벤토리 관리, Q: 나가기): B

현재 보유 골드: 18

구매할 아이템 번호를 입력하세요 (0: 취소): 1

체력 포션을(를) 10 골드에 구매했습니다.

체력 포션이(가) 인벤토리에 추가되었습니다.

10 골드가 인벤토리에서 차감되었습니다. (남은 골드: 8)

테스트 케이스

테스트 제목	테스트 절차	케이스 분류	필수 / 도전	결과
캐릭터 생성 테스트	1. 게임 시작시 캐릭터 이름 입력. 2. 캐릭터 생성 확인	Normal	필수	○
캐릭터 이름 공백 입력	1. 캐릭터 이름을 공백으로 입력.	Abnormal	필수	✕
캐릭터 초기상태 확인	1. 캐릭터 생성 직후 초기 상태 확인.	Normal	필수	○
캐릭터 스탯창 확인	1. 캐릭터 스탯이 생성화면에서 결정했던 수치와 같은지 확인. 2. 스탯 저장기능 확인.	Normal	도전	○
캐릭터 경험치 총량 변동 확인	1. 캐릭터 레벨업 후 경험치 총량 변동되는지 확인.	Normal	도전	○
캐릭터 레벨업 시 능력치 확인	1. 캐릭터 레벨업 후 캐릭터의 능력치가 변동 되는지 확인.	Normal	필수	○
전투 시스템 테스트	1. 전투레벨에 맞는 몬스터 수. 2. 전투레벨마다 랜덤으로 몬스터가 출몰하는지 확인. 3. 공격 및 피격시 HP감소 확인.	Normal	필수	○
전투 골드 및 경험치 드랍 확인	1. 몬스터의 종류마다 서로다른 경험치와 골드를 드랍하는지 확인.	Normal	필수	○
전투 중 사망시 게임종료 확인	1. 캐릭터 체력이 0일 경우 게임 종료.	Normal	필수	○

테스트 케이스

테스트 제목	테스트 절차	케이스 분류	필수 / 도전	결과
캐릭터 특수 능력치 확인.	1. STR, DEX, INT, LUK에 따라 각 능력이 올라갔는지 확인.	Normal	도전	○
캐릭터 스킬 획득 확인.	1. 3레벨업 마다 스킬을 랜덤으로 정상적으로 획득하는지 확인.	Normal	도전	○
전투 중 스킬 사용 확인.	1. 전투 중 스킬 사용시 관련 스킬 이름이 정상적으로 사용되는지 확인.	Normal	도전	○
전투 중 아이템 사용 확인.	1. 전투 중 아이템 사용시 효과가 정상적으로 적용되는지 확인. 2. 아이템 사용 후 캐릭터 인벤토리 아이템 수 변경되었는지 확인.	Normal	필수	○
렌더링 데이터 변경 확인	1. 렌더링 할 오브젝트에 데이터가 잘 주입 되는지 확인 2. 오브젝트에 주입된 데이터가 업데이트 되는지 확인	Normal	도전	○
입력 칸 제한 확인	1. 입력 칸에서 키보드 입력을 받을 때 문자열 길이 제한 확인	Normal	도전	○
로그 출력 확인	1. 로그 출력 문자가 지정된 범위를 벗어나지 않는지 확인 2. 문자 색상이 지정된 색으로 출력되는지 확인	Normal	도전	○

테스트 케이스

테스트 제목	테스트 절차	케이스 분류	필수 / 도전	결과
상점 초기화 확인	1. 상점이 정상적으로 초기화되는지 확인 2. 아이템 목록이 비어있지 않은지 확인	Normal	도전	○
아이템 구매 기능	1. 아이템 가격만큼 골드를 가진 상황에서 아이템 구매 시 정상 처리되는지 확인 2. 구매 후 플레이어의 골드가 정확히 차감되는지 확인 3. 구매한 아이템이 인벤토리에 추가되는지 확인	Normal	도전	○
예외 상황 구매 확인	1. 골드가 부족할 때 구매 시도 시 실패하는지 확인 2. 잘못된 아이템 번호 입력 시 처리되는지 확인	Normal	도전	○
아이템 판매 기능	1. 보유한 아이템 판매 시 정상 처리되는지 확인 2. 판매 후 플레이어의 골드가 정확히 증가하는지 확인 3. 판매한 아이템이 인벤토리에서 제거되는지 확인	Normal	도전	○
예외 상황 판매 확인	1. 인벤토리에서 보유하고 있지 않은 아이템 판매 시도 시 실패하는지 확인 2. 잘못된 아이템 번호 입력 시 처리되는지 확인	Normal	도전	○
인벤토리 표시 확인	1. 플레이어의 인벤토리가 정상적으로 표시되는지 확인 2. 아이템 수량과 골드가 정확하게 표시되는지 확인	Normal	도전	○

필수기능

- ✓ 캐릭터 생성 및 관리
- ✓ 레벨업 시스템
- ✓ 전투 시스템
- ✓ 몬스터 생성
- ✓ 아이템 / 골드 시스템
- ✓ 게임로그

도전기능

- ✓ 보스전 시스템
- ✓ 상점 시스템
- ✗ 인벤토리 관리
- ✓ 아이템 관리

발표 들어주셔서 감사합니다!

Thank you!



Clear the GAME





