

# C++ Text RPG

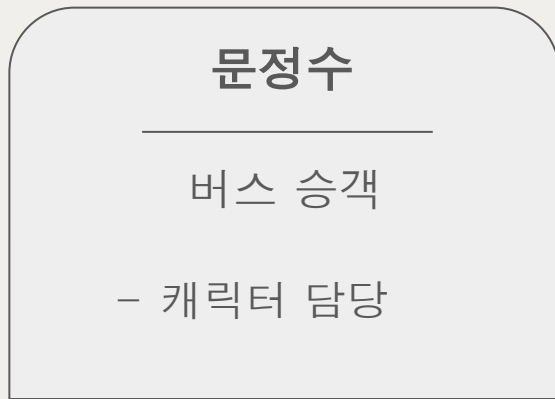
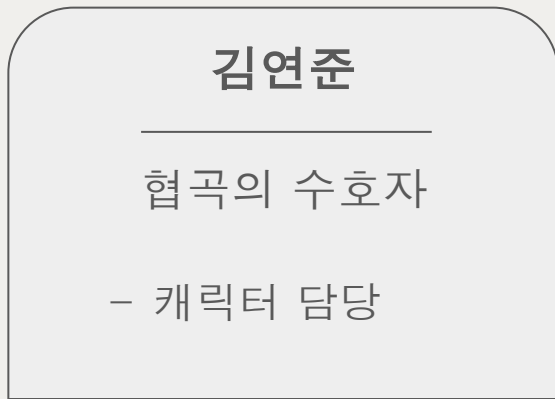
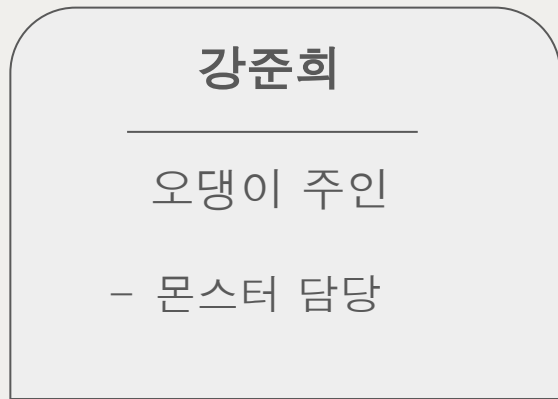
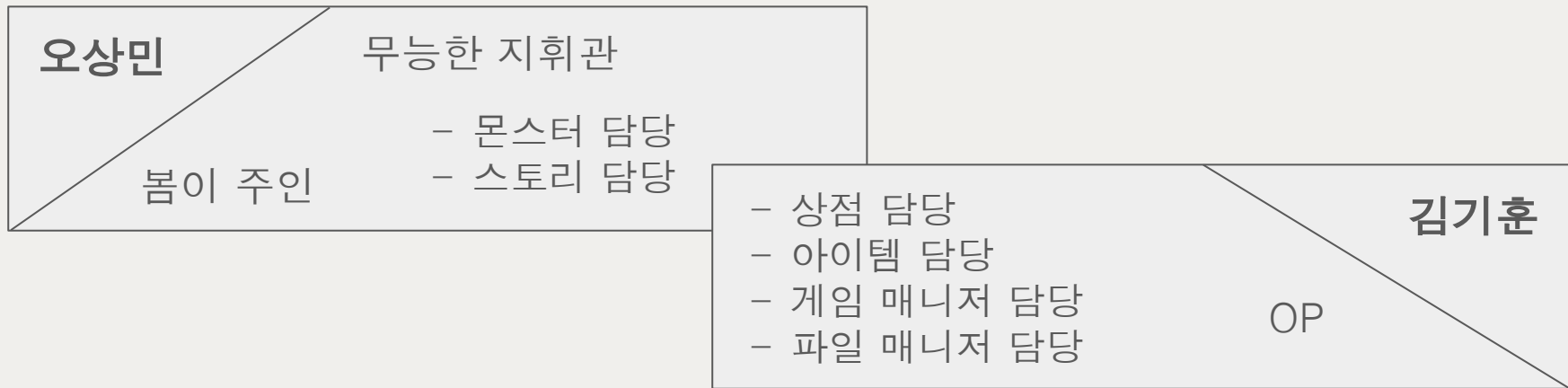
---

텍스트 기반 RPG 게임 기능 구현



오맹이를 봄 조 (10조)

# 오맹이를 봄 조원 소개



1  
.

## Game Flow

간단한 게임 진행 흐름 소개

2  
.

## 주요 클래스 소개 (1)

Monster, Character, Item 클래스

3  
.

## 주요 클래스 소개 (2)

GameManager, FileReaderManager, Store 클래스

4  
.

## 게임 진행 흐름

아이템 효과와 거래 시스템 구현

5  
.

## 시연 영상

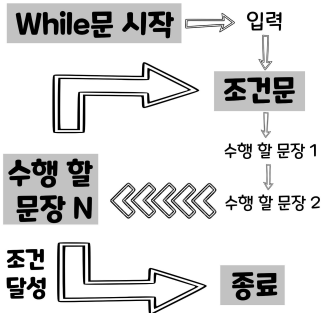
필수, 도전, 추가 기능 시연

# Game Flow



## 스토리

게임 시작 전 스토리 출력  
타이핑 이펙트



## 메인 루프

While 루프  
자동 전투 진행



## 입력 처리

Enter 키를 통한 턴 진행  
번호 입력을 통한 선택지

```
#pragma region Game Logic  
  
> void GameManager::Init() { ... }  
  
> void GameManager::Progress() { ... }  
  
> bool GameManager::Update() { ... }  
  
> void GameManager::Destroy() { ... }  
  
#pragma endregion
```

## 게임 상태

몬스터 사망 시 보상 수령  
캐릭터 사망 시 게임 종료

# 주요 클래스 소개 (1)

싱글톤

주요 함수

## Monster 클래스

- 속성 (캐릭터의 레벨에 따라 변경)
  - 체력(Health)
  - 공격력(Attack)
  - 경험치(Exp)
- 행동
  - 공격, 체력 감소
- 하위 클래스
  - Wolf
  - Goblin
  - Orc
  - Troll
  - Boss
- 데코레이터 패턴
  - 기존 몬스터 칭호 부여
  - 기존 몬스터 추가 능력치 부여

## Character 클래스

- 싱글톤 패턴
- 속성
  - 체력, 최대 체력
  - 레벨
  - 경험치, 레벨 업 필요 경험치
  - 공격력
  - 보유 골드
  - 인벤토리
- 행동
  - 피격
  - 아이템 사용
  - 인벤토리 확인

## Item 클래스

- 멤버 변수
  - int quantity
  - int price
  - string name
- 멤버 함수
  - string getName() = 0;
  - int getPrice() = 0;
  - void use(Character\*);
  - int getQuantity();
- 하위 클래스
  - HealthPotion
  - StrangePotion

## 주요 클래스 소개 (2)

싱글톤

주요 함수

### GameManager 클래스

- 싱글톤 패턴
- 멤버 변수
  - static GameManager\* instance;
- 멤버 함수
  - static GameManager\* **GetInstance()**;
  - static void **DestroyInstance()**;
  - void **Init()**;
  - void **Progress()**;
  - bool **Update()**;
  - void **Destroy()**;
  - static void **setConsoleColor(int)**;
  - int SelectOption();
  - void SelectStore();
  - void SelectStatus();
  - void SelectInventory();
  - Monster\* SelectBattle();
  - void BattleReward(Monster\*);
  - bool SelectEndGame();

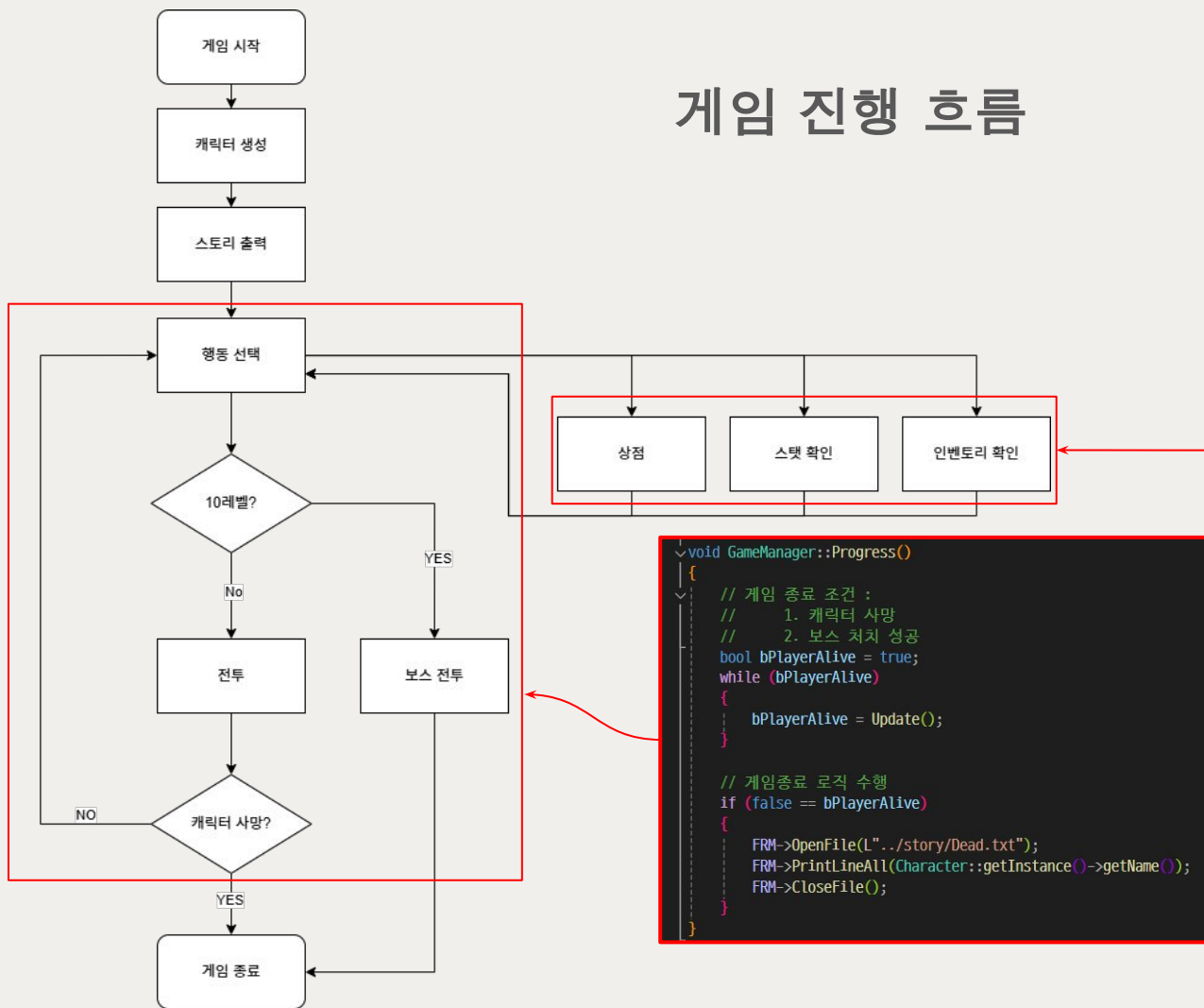
### FileReaderManager 클래스

- 싱글톤 패턴
- 멤버 변수
  - static FileReaderManager\* instance;
- 멤버 함수
  - static FileReaderManager\* **GetInstance()**;
  - static void **DestroyInstance()**;
  - void **ReadFile()**;
  - bool **OpenFile(std::wstring)**;
  - void **PrintLine(int, wstring, bool)**;
  - void **PrintLineAll(wstring, bool)**;
  - void CloseFile();
  - void **typingEffect(const wstring&, int)**;
  - void insertPlayerName(wstring);

### Store 클래스

- 싱글톤 패턴
- 멤버 변수
  - static Store\* instance;
- 멤버 함수
  - static Store\* **getInstance()**;
  - void **destroyInstance()**;
  - void **buyStuff(int)**;
  - void **sellStuff(int)**;
  - void showList();

# 게임 진행 흐름



```

void GameManager::Progress()
{
    // 게임 종료 조건 :
    //   1. 캐릭터 사망
    //   2. 보스 처치 성공
    bool bPlayerAlive = true;
    while (bPlayerAlive)
    {
        bPlayerAlive = Update();
    }

    // 게임종료 로직 수행
    if (false == bPlayerAlive)
    {
        FRM->OpenFile(L"..\\story\\Dead.txt");
        FRM->PrintLineAll(Character::getInstance()->getName());
        FRM->CloseFile();
    }
}
  
```

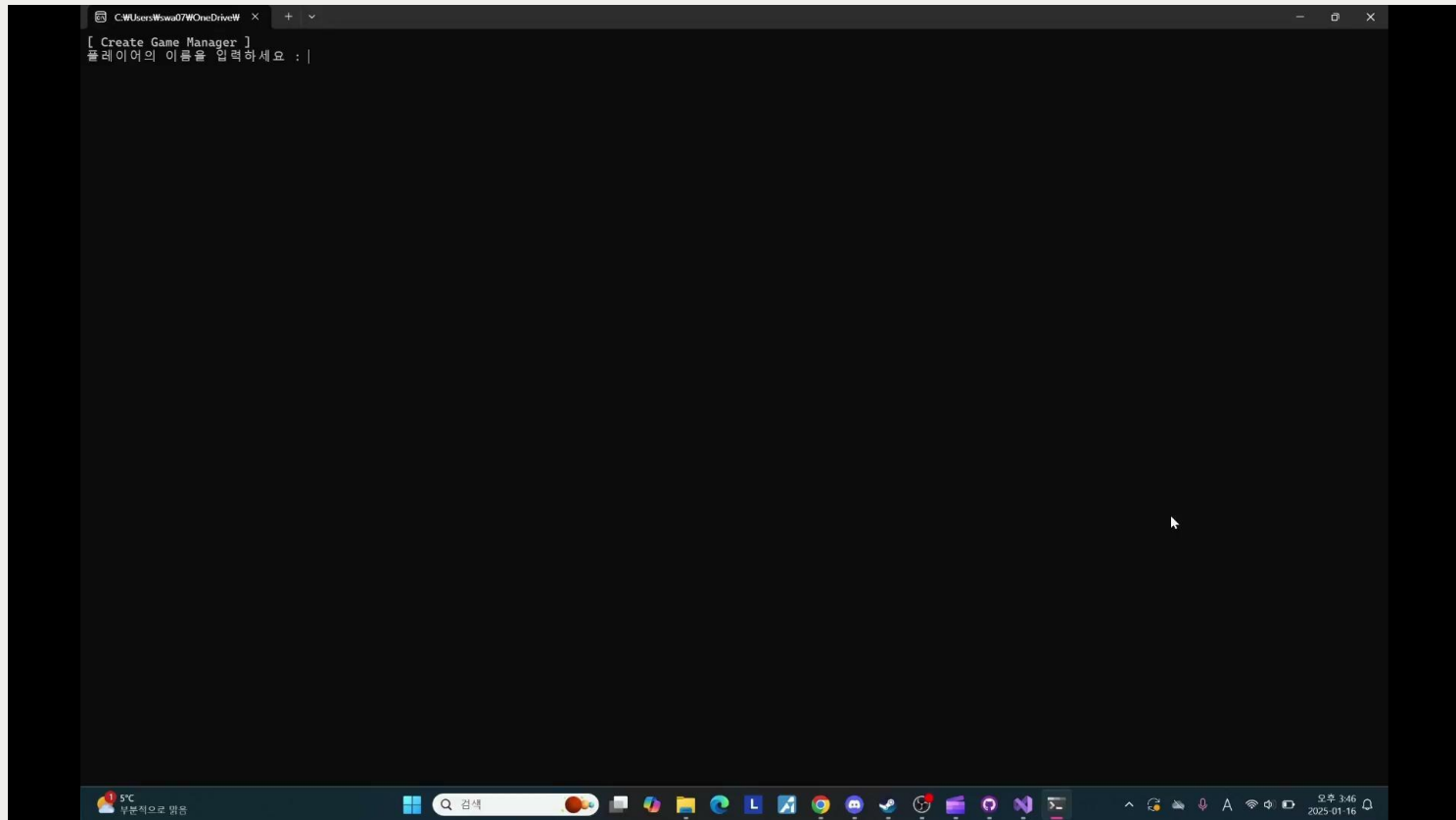
```

bool GameManager::Update()
{
    Character* player = Character::getInstance();
    Store* store = Store::getInstance();
    Monster* genMonster = nullptr;

    // 플레이어 입력
    int Select = SelectOption();
    switch (Select)
    {
    case 1: // 상점
        SelectStore();
        break;
    case 2: // 상태 확인
        SelectStatus();
        break;
    case 3: // 인벤토리 확인
        SelectInventory();
        break;
    case 4: // 전투
        BattleReward(SelectBattle());
        break;
    case 9: // 게임 종료
        return SelectEndGame();
    default:
        __noop;
    }

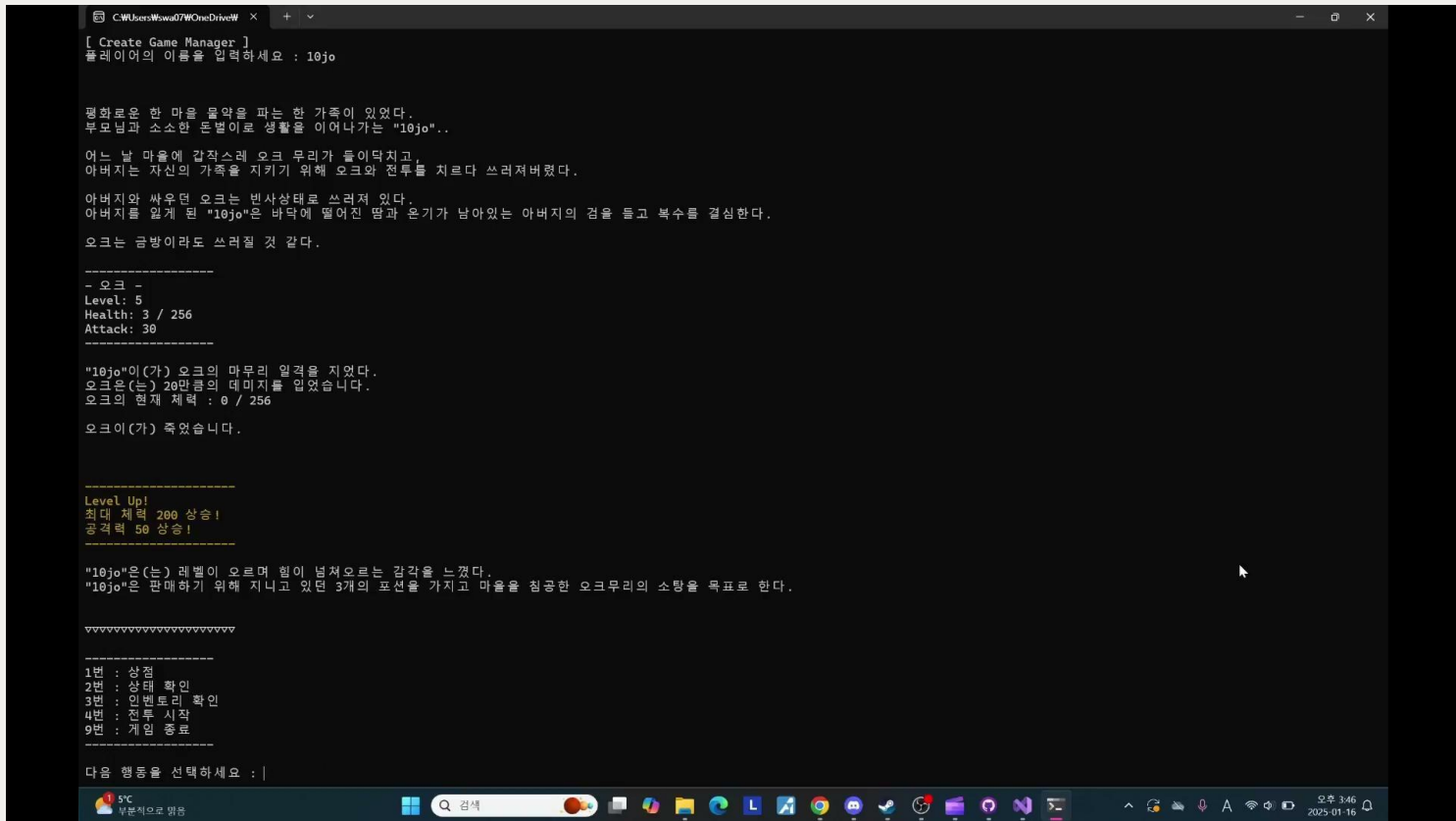
    return player->IsAlive();
}
  
```

# 시연 영상 ( 인트로 )

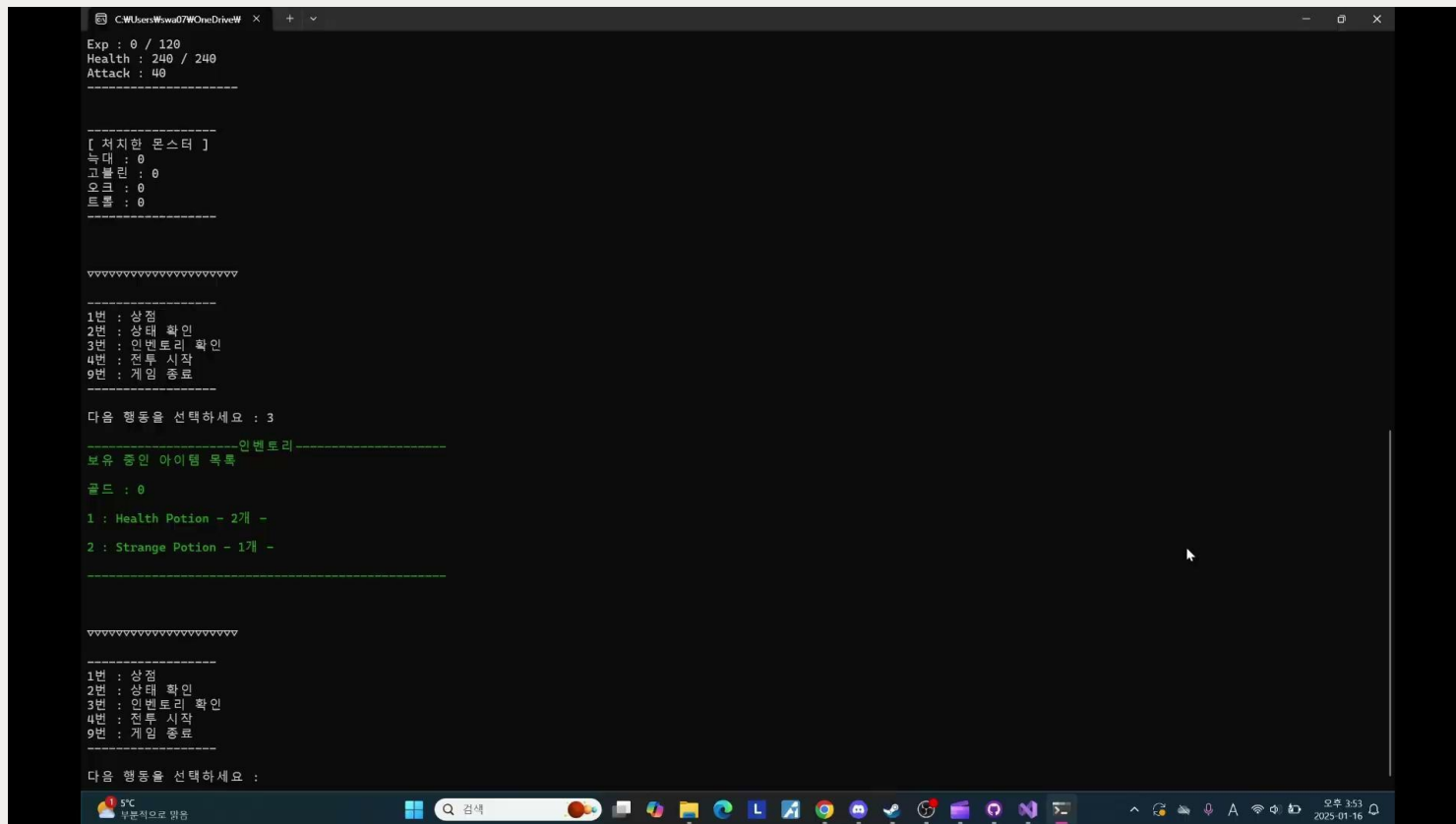




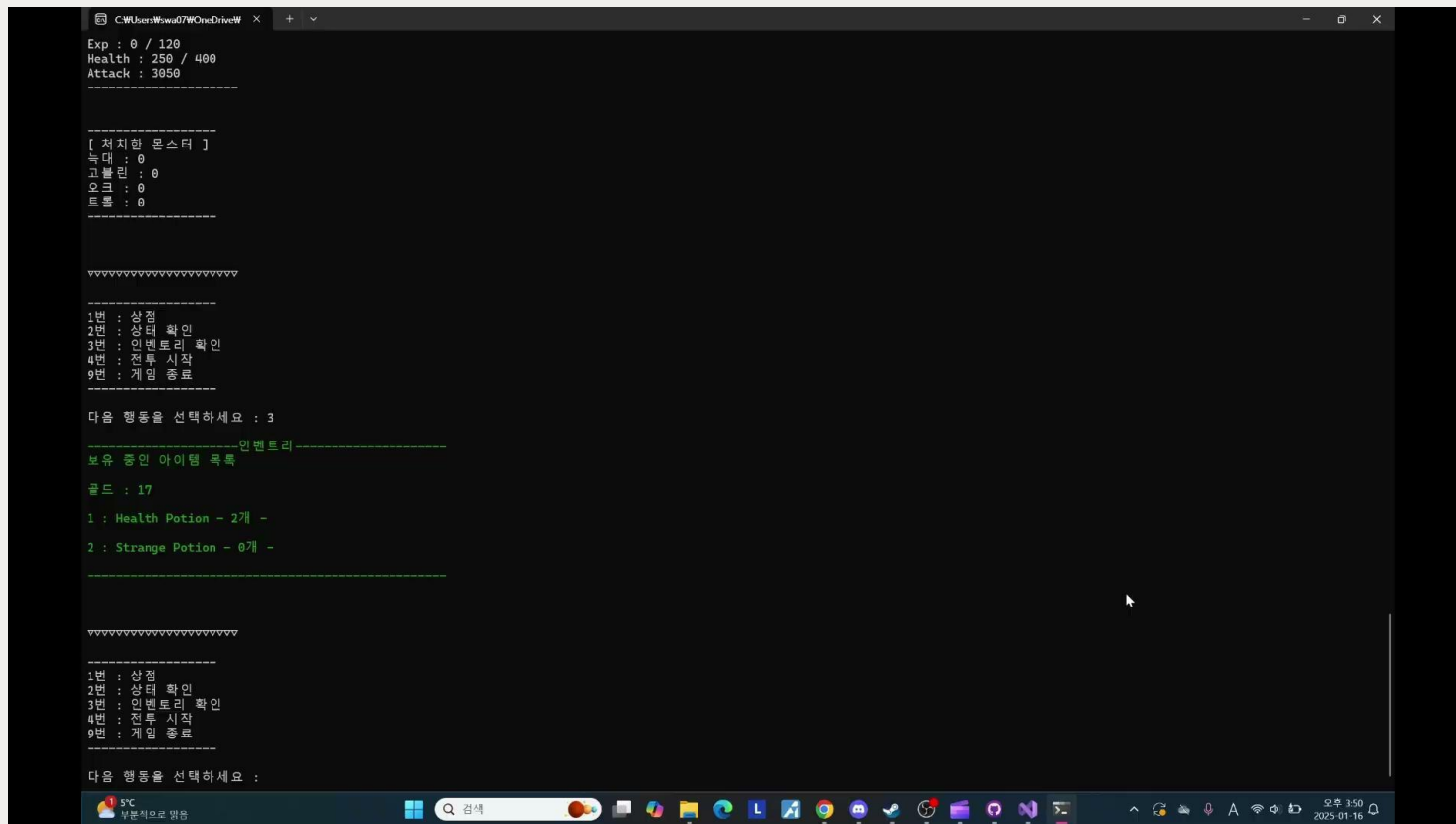
# 시연 영상 ( 상태, 인벤토리 확인 )



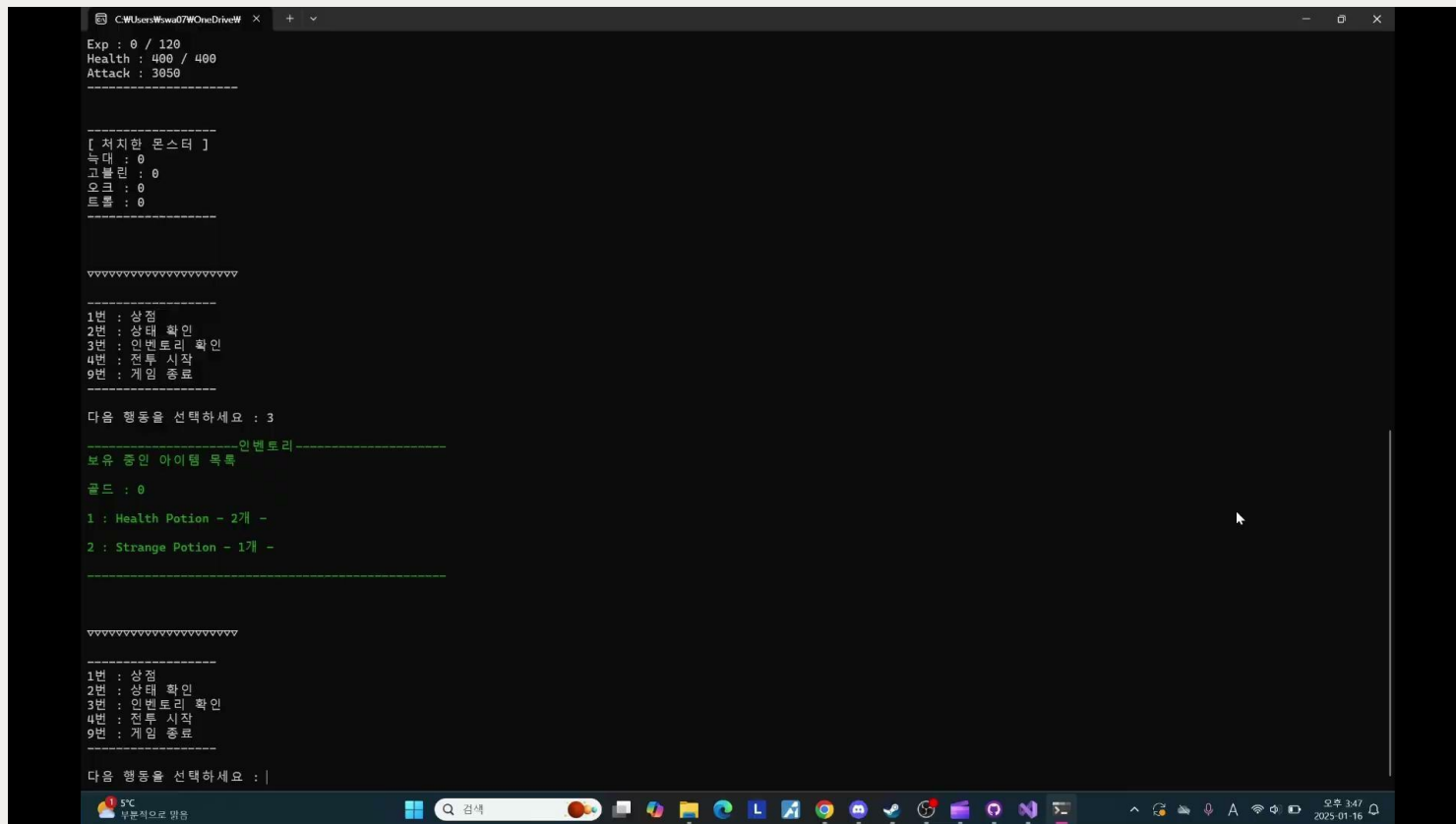
# 시연 영상 ( 전투 )



# 시연 영상 ( 상점 )



## 시연 영상 ( 보스전 )



Thank you

;)