

# 객체지향 프로그래밍 프로젝트

프로젝트 명	기말고사에서 살아남기
팀 명	Protocol 004A
문서 제목	객체지향 프로그래밍 팀프로젝트 최종 보고서

Date	2025 / 05 / 31
------	----------------

팀원	임기범 (5779361)
	장주환 (5779392)
	김지운 (5881270)
	이동우 (5881470)

# 목 차

1	개요 .....	3
1.1	프로젝트 소개 .....	3
1.2	역할 분담 .....	3
1.3	개발환경 .....	3
1.4	개발 기간 .....	3
1.5	개발 목적 .....	3
1.6	개발 목표 .....	4
2	기획 및 설계 .....	4
2.1	게임 장르 및 설명 .....	4
2.2	게임 배경 .....	4
2.3	게임 규칙 .....	4
2.4	엔딩 조건 .....	5
3	시스템 설계 .....	5
3.1	주요 클래스 .....	5
3.1.1	Player 클래스 .....	5
3.1.2	Boss 클래스 .....	5
3.1.3	Item 클래스 .....	5
3.1.4	Skill 클래스 .....	5
3.1.5	BattleSystem 클래스 .....	5
3.1.6	StorySystem 클래스 .....	5
3.1.7	Map 클래스 .....	6
3.1.8	Controller 클래스 .....	6
3.2	아이템 시스템 .....	6
3.3	학점 및 엔딩 시스템 .....	6
3.4	주요 기능 요약 .....	6
4	시연 영상 및 실행 결과 .....	7
5	자기 평가 .....	7
6	결과물 목록 .....	9
7	부록 .....	10
7.1	Github Repository .....	10
7.2	설치 방법 .....	10
7.2.1	설치 방법 1 – git clone .....	10
7.2.2	설치 방법 2 – Download ZIP .....	11
7.3	사용자 매뉴얼 .....	11

# 1 개요

## 1.1 프로젝트 소개

본 프로젝트는 계명대학교 게임 소프트웨어학과와 객체지향 프로그래밍 과목 팀 프로젝트로, C++ 프로그래밍 언어를 기반으로 객체지향 개념(클래스, 상속, 다형성 등)을 활용하여 텍스트 기반 턴 제 RPG 게임을 구현하였습니다. 플레이어와 보스, 중간보스 간의 전투, 아이템 구매/사용, 그리고 학점 시스템을 포함하여 게임의 흐름을 객체지향적으로 구성하였습니다.

## 1.2 역할 분담

- 임기범 : 리더
- 장주환 : 서기 / 기록자
- 김지운 : 총괄 프로그래머
- 이동우 : QA / 테스터

## 1.3 개발환경

- 개발도구 (IDE) : Microsoft Visual Studio 2022
- 운영 체제 (OS) : Window 10 / Window 11
- 프로그래밍 언어 : C++ (표준 : C++17)
- 컴파일러 : MSVS (Microsoft Visual C++)
- 빌드 도구 : Visual Studio 내장 빌드 시스템
- 프로젝트 형식 : 콘솔 애플리케이션 (텍스트기반)
- 기타 툴 : GitHub, GitHub Desktop, Visual Studio code

## 1.4 개발 기간

- 3월 26일 ~ 4월 1일 : 팀 규정문 작성
- 4월 2일 ~ 4월 9일 : 프로젝트 주제 선정 및 역할 분담
- 4월 16일 ~ 5월 30일 : 기능 개발 및 클래스 설계
- 5월 30일 ~ 6월 3일 : 디버깅 및 기능 개선 / 발표 자료 및 보고서 작성

## 1.5 개발 목적

본 프로젝트의 개발 목적은 객체지향 프로그래밍의 주요 개념을 실제 프로젝트의 적용하고, 게임 소프트웨어학부로서 실전 개발 경험을 쌓는 것에 있습니다.

특히 C++ 프로그래밍 언어의 클래스, 상속, 다형성 등의 특징을 활용하여 게임 시스템 전반을 설계하고, 구현하는 능력을 향상시키는 것을 목표로 합니다.

또한 시험과 학점을 소재로 한 텍스트 기반 RPG 게임을 제작함으로써, 학업 상황을 유쾌하게 패러디하고, 사용자가 직접 플레이하며 게임 시스템 기획 및 구현에 대한 이해를 높일 수 있도록 설계하였습니다.

## 1.6 개발 목표

- **C++ 기반 텍스트 기반 RPG 게임 개발**  
콘솔 환경에서 실행되는 게임을 직접 개발하며, 기본적인 게임 구조를 이해하고 구현할 수 있도록 한다.
- **객체지향 개념 실전 적용**  
클래스 구조를 명확히 분리하고, 상속과 다형성을 이용하여 다양한 클래스의 구조를 모듈화 한다.
- **전투 및 스토리 시스템 구현**  
플레이어가 보스와 싸우며 스토리를 진행하고, 이에 따른 엔딩을 보여주는 전투 및 학점 엔딩 시스템을 완성한다.
- **게임 기능 다양화**  
아이템 구매/사용, 공격력/방어력 증가, 학점 평가, 체력 회복 등 다양한 시스템을 이용해 단순한 텍스트 게임을 넘는 흥미요소를 구현한다.
- **학점 기반 멀티 엔딩 구현**  
플레이 결과에 따라 여러 갈래로 나뉘는 엔딩을 사용자에게 제공함으로써 사용자의 선택과 결과에 따른 게임 흐름을 경험할 수 있게 한다.
- **개발 문서 및 발표자료 작성**  
최종 보고서 및 프레젠테이션 슬라이드를 작성하며, 게임기획, 구현, 테스트, 결과까지 전체적인 개발 프로세스를 설명한다.

## 2 기획 및 설계

### 2.1 게임 장르 및 설명

- 텍스트 기반 콘솔 RPG
- 회 차 진행형 스테이지 전투 시스템
- 턴 제 전투 방식

### 2.2 게임 배경

플레이어는 대학생이며, 다니는 학교는 현재 기말고사 기간입니다. 게임은 마지막 날 3 과목 시험을 치르는 상황을 배경으로 하고 있으며, 각 과목의 시험은 "보스전"으로 구현되어 있습니다.

### 2.3 게임 규칙

- 이동 : WASD 키
- 목표 : 3 명의 보스를 이기고 학점 획득

- 학점에 따라 다양한 엔딩이 존재 (해피/배드/노말)

## 2.4 엔딩 조건

맵에 있는 3 개의 B(최종보스)를 모두 클리어 시 엔딩이 나타난다.

# 3 시스템 설계

## 3.1 주요 클래스

본 프로젝트는 객체지향 개념에 따라 각 기능별 클래스를 분리하여 설계 및 구현하였습니다.

아래는 게임에서 핵심 역할을 수행하는 주요 클래스들 입니다.

### 3.1.1 Player 클래스

플레이어의 정보를 관리하며 체력, 공격력, 보유 아이템, 몬스터 아이템, 이름 등의 속성과 관련 메서드를 포함합니다. 게임 내 사용자 캐릭터를 담당합니다.

- 주요 기능 : 체력 회복, 아이템 사용, 공격 정보 제공

### 3.1.2 Boss 클래스

전투에서 플레이어가 상대하는 보스를 정의합니다. 각 보스는 이름, 체력, 공격력, 방어력 등의 속성을 가집니다.

- 주요 기능 : 전투 시 공격, 체력 상태 확인

### 3.1.3 Item 클래스

공격력 증가 또는 체력 회복과 같은 효과를 가진 아이템을 정의합니다. 이름과 효과 수치 등의 속성을 포함합니다

- 주요 기능 : 아이템 효과 반환

### 3.1.4 Skill 클래스

전투에서 사용할 수 있는 스킬을 정의합니다. 스킬은 고유한 이름, 데미지, 성공확률을 가집니다.

- 주요 기능 : 스킬 정보 제공

### 3.1.5 BattleSystem 클래스

플레이어와 보스 간의 전투를 처리하는 전투 시스템을 담당합니다. 각 보스와의 전투 흐름과 학점 산정 로직이 포함되어 있습니다.

- 주요 기능 : 공격 처리, 체력 관리, 학점 평가, 전투 종료 처리

### 3.1.6 StorySystem 클래스

게임의 서사를 구성하며, 각 전투 결과에 따라 다양한 스토리 라인을 출력하고, 학점 평균에 따른 엔딩을 출력합니다.

- 주요 기능 : 게임 시작 이야기 출력, 전투 후 결과 출력, 엔딩 분기 처리

### 3.1.7 Map 클래스

게임의 맵을 구성하며 플레이어의 이동, 상태 출력, 상점 진입, 전투 시작등을 담당합니다.

- 주요 기능 : 키 입력 처리, 전투 트리거 등

### 3.1.8 Controller 클래스

게임의 전체 흐름을 제어하는 메인 클래스입니다. 초기화, 게임 루프 실행, 시스템 간의 연결 등을 담당합니다.

- 주요 기능 : 게임 루프 시작, Map 클래스 호출, 프로그램 종료 처리 등

## 3.2 아이템 시스템

플레이어는 총 1 개의 아이템을 사용할 수 있으며, 다음과 같은 효과를 갖습니다.

1. 전공책 : 공격력 120% 증가
2. 계산기 : 공격력 110% 증가
3. 휴대폰 : 공격력 130% 증가
4. 몬스터 : 체력 30% 회복

1 ~ 3 번은 중복 구매 불가, 4 번은 반복 사용 가능

## 3.3 학점 및 엔딩 시스템

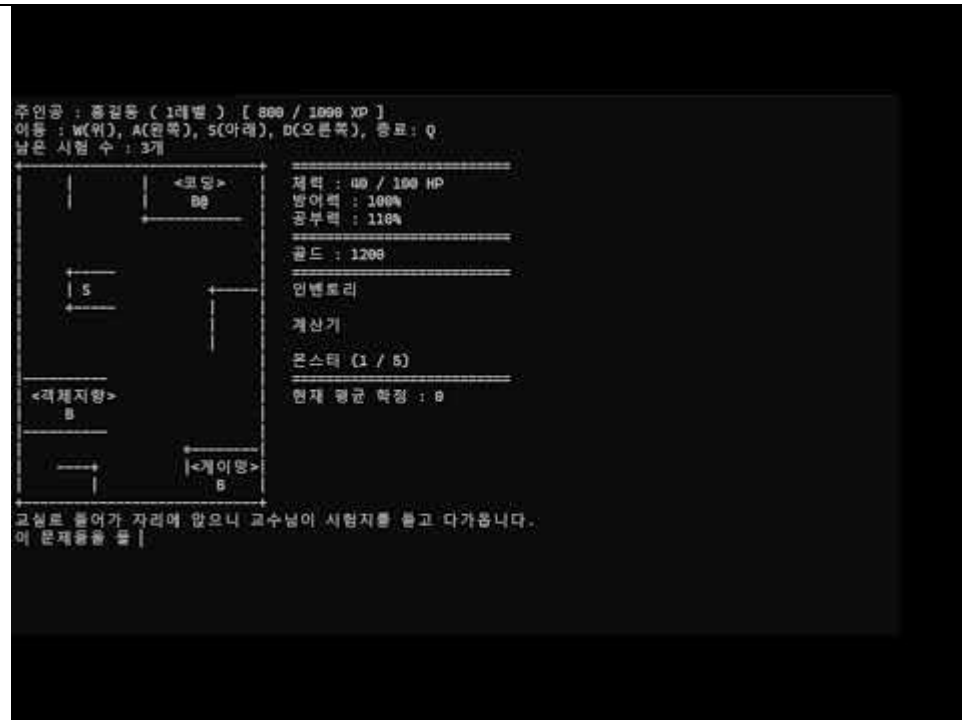
- 각 보스를 클리어 시 플레이어 체력에 따라 A(4.0, B(3.0), C(2.0) 학점 부여
- 보스에게 패배시 F(0.0) 학점 부여
- 학점 평균 계산 후 다음과 같은 엔딩 분기

평균 학점	엔딩 유형	엔딩 설명
4.0	해피엔딩	완벽한 기말고사!
3.0 이상	엔딩 1	무난한 기말고사
2.0 이상	엔딩 2	평범하게 못 친 기말고사
2.0 미만	배드엔딩	학사경고 3 회, 제적통보

## 3.4 주요 기능 요약

- 아이템 사용
- 스킬 선택과 전투 진행
- 학점 저장 및 평균 계산
- 학점에 의해 엔딩 출력

## 4 시연 영상 및 실행 결과



<https://www.youtube.com/watch?v=NLtkak2RMYS>

## 5 자기 평가

팀원	자기평가
임기범	본 프로젝트에서 상점 시스템, 엔딩 시스템, 디버깅을 담당하였고, C++과 객체지향 개념 적용에 익숙하지 않아 많은 어려움을 겪었습니다. 기존의 절차지향 방식과 달리 객체지향 설계는 처음엔 낯설었지만, 사용하면서 유지보수성과 확장성 면에서 장점을 체감할 수 있었습니다. 팀 프로젝트 경험도 처음이었지만, 팀원들과의 협업을 통해 역할 분담과 개발을 성공적으로 마칠 수 있었습니다. 다소 아쉬운 부분도 있었지만, 전투, 아이템, 스토리 시스템을 직접 구현하며 게임 개발의 구조를 경험했고, GitHub를 통한 협업 또한 큰 배움이 되었습니다.
장주환	본 프로젝트에서 게임 로직 및 전투 시스템 구현을 담당하며 핵심 매커니즘을 구현하고, 회의록 작성을 작성하였습니다. 개발 중 사소한 오류 및 버그가 빈번하게 발생했으나,

	<p>팀원과의 협력을 통해 해결하였습니다. 콘솔 기반 환경으로 인해 시각적인 몰입도가 다소 제한적이었으나, 개선점을 발견하여 향후 발전 가능성을 확인했습니다. 이번 프로젝트로 김 협업의 중요성을 깨닫고 객체지향 설계 및 디버깅 능력을 향상시켰습니다.</p>
김지운	<p>텍스트 기반 RPG 게임에서 저는 개발자로서 전체적인 게임 흐름을 설계하고, 캐릭터 스탯 시스템, 전투 로직, 맵 이동 기능 등을 C++로 구현하였습니다.</p> <p>가장 어려웠던 점은 전투 시스템의 흐름 정리였습니다</p> <p>처음에는 공격 결과, HP 변화, 스킬 사용 등의 메시지가 복잡하게 출력되어 플레이어가 혼란을 겪었습니다. 이를 해결하기 위해 출력 순서를 정리하고, 메시지를 간결하게 표시하는 방식으로 수정하였습니다.</p> <p>C++ 구조체와 클래스를 통해 객체 지향적으로 게임을 구성하는 방법을 익혔으며 사용자 입력 처리, 예외 처리, 전반적인 게임 시스템 설계 능력이 향상되었습니다.</p> <p>인터페이스가 텍스트 기반이라 시각적 흥미가 떨어졌던 점이 아쉬웠습니다. 추후에는 콘솔 그래픽(색상 출력, 간단한 애니메이션 등)이나 GUI 라이브러리를 활용해 개선하고 싶습니다.</p> <p>이번 텍스트 RPG 게임 제작은 프로그래밍 실력을 실제 프로젝트에 적용해볼 수 있는 매우 뜻깊은 경험이었습니다. 단순히 코드를 작성하는 것을 넘어, 전체 게임의 흐름을 고민하고 플레이어의 입장에서 생각해보는 과정에서 게임 기획과 개발에 대한 통합적 사고 능력이 향상되었습니다. 다음에는 좀 더 발전된 형태의 게임을 만들어보고 싶습니다.</p>
이동우	<p>이번 객체지향 텍스트 게임 프로젝트에서 저는 게임의 틀에 대한 아이디어, 부족한 부분에 대한 의견 개진을 맡았습니다.</p> <p>처음에는 클래스 간의 관계를 어떻게 나누고 연결할지 설계하는 것이 어려웠습니다. 역할이 겹치거나 책임이 명확하지 않은 클래스가</p>



	<p>생겨 유지보수가 어려웠기 때문입니다. 이를 해결하기 위해 UML 다이어그램을 그려보며 각 클래스의 책임을 재정의하고, 기능을 단순화하는 리팩토링을 반복하였습니다. 또한 전투 시스템에서 랜덤성과 사용자 선택을 동시에 처리하는 부분에서 버그가 발생했는데, 디버깅을 통해 조건문과 입력 처리 부분을 꼼꼼히 확인하고 수정하였습니다.</p> <p>이번 프로젝트를 통해 객체지향 프로그래밍의 기본 개념인 캡슐화, 상속, 다형성을 실제로 적용해보며 그 중요성과 활용 방법을 깊이 이해할 수 있었습니다. 또한, 텍스트 기반이지만 사용자와의 상호작용을 고려한 로직 설계의 중요성을 배웠고, 유지보수가 쉬운 구조를 만드는 데 있어서 명확한 책임 분리가 얼마나 중요한지도 느꼈습니다.</p> <p>아쉬운 점은 게임의 UI가 텍스트 기반이다 보니 사용자 경험이 다소 단조로웠다는 점입니다. 또한, 클래스 간의 의존성을 완전히 분리하지 못한 부분이 있어 추후 기능 확장 시 제약이 있을 수 있다고 느꼈습니다.</p> <p>다음에는 디자인 패턴을 활용해 좀 더 유연하고 확장성 있는 구조를 만들고, 간단한 GUI라도 적용해 사용자 인터페이스를 개선하고 싶습니다.</p> <p>객체지향 개념을 실제 프로젝트에 적용해보며 많은 것을 배우고 성장할 수 있는 시간이었다고 생각합니다. 처음에는 막막했지만, 점점 구조를 잡아가며 내가 만든 게임이 동작하는 모습을 볼 때 큰 보람을 느꼈습니다. 특히, 단순한 텍스트 게임이라도 객체지향적으로 잘 설계하면 재사용성과 유지보수성이 높아진다는 것을 체감할 수 있었습니다. 앞으로 더 복잡한 프로그램도 자신 있게 도전할 수 있을 것 같습니다.</p>
--	--

## 6 결과물 목록

파일명	역할
-----	----

BattleSystem.cpp BattleSystem.h	플레이어와 보스 간의 전투를 처리하는 시스템. 공격 선택, 데미지 계산, 학점 계산 등을 포함
Boss.cpp Boss.h	보스 몬스터 정보 관리. 이름, 체력, 공격력 등을 포함하며 전투 대상 보스를 정의함.
Character.cpp Character.h	Player 와 Boss 가 공통으로 가시는 속성을 정의. 죽음 감지, 데미지 감지 등을 선언.
Controller.cpp Controller.h	게임의 전체 실행 흐름을 담당. 맵 출력, 전투 진입, 종료까지를 통합 관리
Item.cpp Item.h	아이템 객체 정의. 이름, 공격력 증가 수치 등을 포함하며, 전투 중 사용 가능
main.cpp	게임의 시작 흐름을 제어하며, Controller 객체를 실행시켜 전체 시스템을 구동함
Map.cpp Map.h	플레이어의 위치, 이동, 상점, 전투 구역 등을 관리. WASD 키 입력 처리, 상태창 출력 등 포함
Player.cpp Player.h	플레이어 객체의 상태를 관리. 아이템 사용, 공격력 반환, 체력 회복 등을 기능 수행.
Shop.cpp Shop.h	아이템 구매 로직 처리. 상점 시스템 처리 플레이어의 동 차감 및 아이템 추가 수행
Skill.cpp Skill.h	스킬 객체 정의. 스킬 이름, 데미지, 성공 확률 등 전투에 사용되는 기술 정보 관리.
StorySystem.cpp StorySystem.h	게임 스토리 및 엔딩 출력 담당. 학점 평균에 따라 엔딩 분기 출력.

## 7 부록

### 7.1 Github Repository

<https://github.com/baaaaam2/project004A>

### 7.2 설치 방법

#### 7.2.1 설치 방법 1 – git clone

1. 게임을 설치하고자 하는 디렉토리로 이동하기 위해 터미널에 다음과 같이 입력한다.

```
cd 게임을 설치하고자 하는 디렉토리
ex) cd Desktop
```

2. 다음과 같은 명령어를 입력한다.

```
git clone https://github.com/baaaaam2/project004A
```

### 7.2.2 설치 방법 2 – Download ZIP

1. 다음과 같은 주소를 인터넷 브라우저 주소창에 입력한다.

<https://github.com/baaaaam2/project004A/archive/refs/heads/main.zip>

2. 다운로드가 완료되면 파일을 게임을 설치하고자 하는 디렉토리에 압축 해제한다.

## 7.3 사용자 매뉴얼

- **Visual Studio 에서 열기**

1. Visual studio 실행
2. 메뉴 -> 파일 -> 열기 -> 프로젝트/솔루션
3. 압축 해제한 폴더 내의 .sln 파일 선택

- **프로젝트 빌드 및 실행**

- 상단 메뉴 -> 디버그 -> 디버깅하지 않고 시작(Ctrl + F5)
- 또는 F5 로 디버깅 실행