

# 최종 보고서

조원 : 김민준,김민성,여금비,최영환

## 1. 서론

본 프로젝트는 스케줄 기반 성장 게임으로써, 60일동안 각 스탯을 관리하며 프로젝트 진행도를 올리는 게임입니다.

진행도를 올리기 위해서는 4개의 스탯(돈, 진행도, 협력도, 체력)을 잘 분배하여야 하고, 60일이 되었을 때, 진행도에 따라 엔딩이 결정됩니다.

## 2. 시스템 구조 및 모듈 설명

프롤로그: 사용자 이름 입력 및 스토리 도입부 출력

FileIO 유틸리티: 텍스트 파일 입출력 및 문장 단위 처리

상태 및 통계 관리: 사용자 상태 추적 및 통계 분석

시각화 및 메인 실행 흐름: 메인 루프 및 시각적 출력 처리

## 3. 주요 헤더파일 설명

### 1. prolog.h

사용자로부터 이름을 입력받고, 도입부 스토리를 출력합니다.

FileIO::printFile()을 통해 외부 텍스트 파일을 출력하며,

ENTER 입력을 통해 다음 단계로 넘어갑니다.

콘솔 화면을 `system("cls")`로 초기화하여 몰입감을 제공합니다.

## 2. FileIO.h / ProimagIO.h / Event\_imag\_IO.h

세 파일은 거의 동일한 기능을 제공하며, 파일 입출력 유틸리티를 정의합니다.

ProimagIO.h와 Event\_imag\_IO.h 는 교수님의 표정과,

이벤트가 발생했을 시 이미지 출력을 담당하는 헤더입니다.

FileIO.h는 총 4개의 txt file(Part\_time\_job\_story, Playing\_story, Rest\_story, Work\_story) 등을 매개변수에 들어온 index값에 맞게 출력하는 역할을 담당합니다.

FileIO.h 의 주요 함수:

trim: 문자열 공백 제거

readLines: 주석과 공백을 제외한 텍스트 라인 읽기

readSentences: 텍스트를 문장 단위로 분리

printFile: 전체 파일 출력

printSentenceAt: 특정 문장 출력 (UTF-8 인코딩 설정 포함)

## 3. next\_day.h

하루가 끝난 후의 결과(일정, 상태변화)를 사용자에게

시각적으로 보여주는 역할을 하는 헤더입니다.

주요 함수:

print\_day\_result:

하루 동안의 활동 결과를 출력

출력 항목 :

- 날짜 및 D-day
- 오전 오후 일정
- 돈, 체력, 진행도, 팀워크의 변화량과 현재 상태

각 상태는 막대 그래프 형태로 시각화 됩니다.

#### 4. stats.h / stats.cpp

사용자 상태를 추적 및 업데이트 하는 기능을 제공하는 헤더입니다.

주요 전역 변수:

Int health - 체력 (0~100)

Int progress - 진행도 (0~100)

Double money - 자금 (0~ )

Int teamwork - 팀워크 (0~100)

#### 5. GameController.h

게임의 기능을 담당하고 있는 파일입니다. (예: 일과선택, 상점이동, 다음날 이동)

주요 함수:

run: 프롤로그 출력, 날짜에 따른 이벤트 출력, 플레이어의 일정 선택, 일정에 따른 이벤트 출력 등 게임의 주요 기능을 가진 함수입니다.

## 6. store.h

상점의 기능을 담고 있는 헤더파일입니다.

각종 아이템을 구매할 수 있으며, 구매한 아이템은 stats에 있는 함수와 연결되어  
구매한 아이템은 stats을 변화시킵니다.

주요 함수:

print\_store: 상점의 UI를 출력하고 아이템 구매 기능을 담당하는 함수입니다.

## 7. visual\_main.h

사용자에게 하루 일정 선택 UI를 제공하고, 현재 상태(Health, Teamwork, Progress)를 시각적으로 출력하는 역할을 합니다.

주요 함수:

SetMorningPlan(int choice)

SetAftrenoonPlan(int choice)

->받은 choice 값 1(알바하기), 2(놀러가기), 3(과제하기), 4(쉬기) 에 따라서 일과를 설정할 수 있도록 도와준다.

## 8. ending\_image.h

게임의 엔딩에 출력되는 아스키아트를 u8R"( )"로 문자열 출력방식을 UTF-8 형식으로 변경하여 아스키아트를 출력하는 헤더파일 입니다.

## 9. Start\_Up.h

프로그램 시작 시 시작화면(ASCII아트)를 출력하는 역할을 합니다.

\_getch()를 통해 사용자가 아무 키나 누를 때까지 대기합니다.

## 4. 주요 기능 설명

1. 사용자 입력 처리 prolog 클래스에서 사용자 이름을 입력받고, 도입부 스토리를 출력 visual\_main 클래스에서 오전/오후 일정 선택을 위한 숫자 입력 처리 \_getch() 및 getline()을 통해 키 입력 및 문자열 입력을 처리
2. 텍스트 기반 스토리 출력 ProimagIO 네임스페이스의 printFile() 및 readSentences() 함수를 통해 외부 텍스트 파일을 줄 또는 문장 단위로 출력 prolog 클래스에서 Story/prolog1.txt, prolog2.txt 파일을 출력하여 스토리 도입부 구성
3. 일정 선택 및 변환 visual\_main 클래스에서 오전/오후 일정 선택을 숫자로 입력 받고, getPlan()을 통해 텍스트로 변환 next\_day 클래스의 convertPlan() 함수는 숫자 문자열을 일정 텍스트로 변환하여 결과 화면에 출력
4. 상태 변화 및 관리 stats 모듈에서 health, progress, money, teamwork 등의 상태를 전역 변수로 관리 initializeStats()로 초기화, updateStats()로 변화 적용 상태값은 범위를 벗어나지 않도록 제한 처리 포함
5. 하루 결과 리포트 next\_day 클래스의 print\_day\_result() 함수는 하루 동안의 일정, 상태 변화, D-Day 등을 종합적으로 출력 콘솔 화면을 초기화하고, 결과를 정리된 형식으로 보여줌
6. 콘솔 기반 UI 구성 visual\_main 클래스에서 날짜, 일정, 상태 등을 포함한 메인 UI 구성 system("cls")를 통해 화면을 전환

## 5. 기술적 구현 세부사항

1. 콘솔 인코딩 및 출력 처리 SetConsoleOutputCP(CP\_UTF8)을 사용하여 콘솔 출력

인코딩을 UTF-8로 설정함으로써, 한글 및 유니코드 아트 출력이 가능하도록 구현 system("cls")를 통해 콘솔 화면을 초기화하여 화면 전환 효과를 제공

2. 상태 관리 (Stats) stats.h와 stats.cpp에서 전역 변수(health, progress, money, teamwork)를 선언 및 정의하여 중앙 집중식 상태 관리 구현 initializeStats(), updateStats(), printStats() 함수로 상태 초기화, 갱신, 시각화 기능 제공 상태값은 0~100 범위로 제한되며, 콘솔에 막대 그래프 형태로 출력
3. 텍스트 기반 UI 구성 visual\_main 클래스에서 하루 일정 선택 UI를 구성 오전/오후 일정 선택 현재 상태 출력 (Health, Teamwork, Progress) next\_day 클래스에서 하루가 끝난 후의 결과를 출력 일정 결과, 상태 변화량, D-Day 등을 포함한 결과 리포트 제공
4. 파일 입출력 유틸리티 ProimagIO.h 및 Partimejob\_imag\_IO.h에서 다음 기능 제공: readLines(): 텍스트 파일을 줄 단위로 읽고 주석 제거 readSentences(): 텍스트를 마침표 기준으로 문장 분리 printFile(), printSentenceAt(): 텍스트 출력 기능 trim() 함수로 문자열 앞뒤 공백 제거
5. 일정 선택 및 변환 일정 선택은 숫자(1~4)로 입력받고, getPlan() 과 convertPlan() 함수를 통해 텍스트로 변환 일정에 따라 상태 변화가 적용되며, updateStats()를 통해 이벤트에 따른 스탯 변화율이 반영됨

## 6. 테스트 및 결과

테스트 시나리오:

이름 입력 → 프롤로그 → 메인화면 → 일과선택 →스토리 출력 → 상태 변화 → 통계 확인 → 엔딩 확인