**C++프로그래밍**

**프로젝트**

|  |  |
| --- | --- |
| **프로젝트 명** | Snake Game |
| **팀 명** | 11조 |
| **문서 제목** | 결과보고서 |

|  |  |
| --- | --- |
| **Version** | 1.3 |
| **Date** | 2024-06-15 |

|  |  |
| --- | --- |
| **팀원** | 박진오(팀장) |
| 김찬중 |
| 조성민 |
|  |
|  |

|  |
| --- |
| **CONFIDENTIALITY/SECURITY WARNING**  이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 C++프로그래밍 수강 학생 중 프로젝트 “Snake Game”를 수행하는 팀 “11조”의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 “11조”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다. |

**문서 정보 / 수정 내역**

|  |  |
| --- | --- |
| **Filename** | 최종보고서\_SnakeGame\_11조.doc |
| **원안작성자** | 박진오, 김찬중, 조성민 |
| **수정작업자** | 박진오, 김찬중, 조성민 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 수정날짜 | 대표수정자 | Revision | 추가/수정 항목 | 내 용 |
| 2024-06-03 | 김찬중 | 1.0 | 최초 작성 | 초안 작성 |
| 2024-06-7 | 조성민 | 1.1 | 내용 수정 | 수정된 연구내용 추가 |
| 2024-06-12 | 박진오 | 1.2 | 내용 수정 | 수정된 연구내용 추가 |
| 2024-06-17 | 박진오 | 1.3 | 최종 수정 | 마무리 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**본 양식은 소프트웨어학부 C++프로그래밍 과목의 프로젝트 결과보고서 작성을 위한 기본 양식입니다. 문서의 필수 항목을 제시하는 것이니 폰트, 문단 구조 등의 디자인 부분은 자유롭게 설정하기 바랍니다. 양식 내에 붉은 색으로 기술한 부분은 지우고 작성하기 바랍니다.**

**목 차**

[1 개요 4](#_gjdgxs)

[2 개발 내용 및 결과물 5](#_30j0zll)

[2.1 목표 5](#_1fob9te)

[2.2 개발 내용 및 결과물 6](#_3znysh7)

[2.2.1 개발 내용 6](#_2et92p0)

[2.2.2 시스템 구조 및 설계도 6](#_tyjcwt)

[2.2.3 활용/개발된 기술 6](#_3dy6vkm)

[2.2.4 현실적 제한 요소 및 그 해결 방안 6](#_1t3h5sf)

[2.2.5 결과물 목록 7](#_2s8eyo1)

[3 자기평가 8](#_17dp8vu)

[4 참고 문헌 8](#_3rdcrjn)

[5 부록 8](#_26in1rg)

[5.1 사용자 매뉴얼 8](#_lnxbz9)

[5.2 설치 방법 8](#_35nkun2)

# **개요**

|  |
| --- |
| **평가기준 (10점)**  **프로젝트를 완성하기 위해 사용한 개발 방법을 기술하세요.**  **또한 사용하고 있는 외부 라이브러리와 해당 라이브러리를 획득/설치하는 방법을 기술하세요.** |

## **프로젝트 소개**

국민대학교 C++프로그래밍 수업의 과제인 Snake Game Project는 C++ 언어를 사용하여 구현되었다. 이 프로젝트에서는 ncurses 라이브러리를 활용하여 터미널 환경에서 실행되는 텍스트 기반의 스네이크 게임을 제작하였다. 주요 목표는 C++ 언어의 기본적인 문법과 데이터 구조, 알고리즘을 활용하여 게임을 설계하고 구현하는 것이다.

**게임의 주요 기능**

1. 뱀의 이동 및 성장

2. 게임 맵 생성 및 업데이트

3. 아이템과 게이트 생성 및 상호작용

4. 점수 계산 및 표시

이 프로젝트를 통해 팀원들은 C++ 프로그래밍 언어의 심화된 기능을 학습하고, ncurses 라이브러리를 이용한 텍스트 기반의 UI 구현 방법을 익히는 것을 목표로 하였다. 또한, 팀 프로젝트를 통해 협업 능력과 문제 해결 능력 키우고자 한다.

## **개발 방법**

개발 방법은 계획 및 설계, 구현, 테스트 및 디버깅, 최적화의 네 단계로 나누어 진행되었다. 각 단계별로 구체적인 내용은 다음과 같다.

**1. 계획 및 설계 단계**

목표 설정: 프로젝트의 주요 목표와 기능을 정의하였다. 이를 위해 게임의 기본 로직과 구조를 설계하고, 필요한 클래스와 함수의 역할을 명확히 하였다.

파일 구조 설계: 전체 프로젝트의 파일 구조를 계획하고, 각 파일의 역할을 분리하였다. main.cpp, Snake.cpp, MapManager.cpp 등의 파일로 나누어 개발을 시작하였다.

**2. 구현 단계**

게임 초기화 및 루프 - main.cpp 파일에서 게임의 초기화, 메인 루프, 종료 처리를 구현하였다. 이 과정에서 ncurses 라이브러리를 초기화하고 종료하는 코드를 작성하였다.

뱀의 이동 및 조작 - Snake.cpp와 Snake.h 파일에서 뱀의 상태와 동작을 관리하는 클래스를 구현하였다. 뱀의 이동, 먹이 먹기, 충돌 감지 등의 기능을 작성하였다.

맵 관리 - MapManager.cpp와 MapManager.h 파일에서 게임 보드의 상태를 관리하는 클래스를 구현하였다. 맵의 생성, 업데이트, 표시 기능을 담당하였다.

**3. 테스트 및 디버깅 단계**

기능별 테스트 - 각 기능을 개별적으로 테스트하여 오류를 발견하고 수정하였다. 예를 들어, 뱀의 이동 기능을 테스트하여 예상치 못한 충돌이나 이동 오류를 해결하였다.

통합 테스트 - 모든 기능을 통합하여 전체 게임이 의도대로 동작하는지 확인하였다. 게임 진행 중 발생하는 버그를 찾아 디버깅하였다.

디버깅 도구 활용 - 디버깅 도구와 로그 출력을 사용하여 오류를 추적하고 수정하였다.

**4. 최적화 단계**

성능 분석 및 최적화 - 코드의 성능을 분석하고, 불필요한 연산이나 메모리 사용을 줄이기 위한 최적화를 진행하였다. 예를 들어, 뱀의 이동 시 불필요한 데이터 복사를 줄이는 등의 작업을 하였다.

코드 리뷰 및 개선 - 주기적인 코드 리뷰를 통해 코드의 가독성과 유지보수성을 개선하였다. 팀원 간 코드 리뷰를 통해 발견된 문제점을 수정하고 최적의 해결책을 모색하였다

## **사용 라이브러리**

**1. NCURSES**

**역할:** 텍스트 기반의 사용자 인터페이스를 제공하는 라이브러리이다. 이 프로젝트에서는 게임 화면을 그리거나, 사용자의 입력을 처리하는 데 사용되었다.

Ubuntu 터미널에 다음 명령어를 입력하여 설치할 수 있었다.

: sudo apt-get update

: sudo apt-get install libncurses5-dev libncursesw5-dev

**2. STL (Standard Template Library)**

**역할**: C++ 표준 라이브러리로, 데이터 구조와 알고리즘을 제공한다. 이 프로젝트에서는 주로 vector, deque, map 등의 컨테이너를 사용하였다.

**설치 방법:** STL은 C++ 표준 라이브러리의 일부로, 별도의 설치 과정이 필요 없다. C++ 컴파일러(g++, clang++)를 설치하면 자동으로 포함된다.

## **설치 방법**

Ubuntu : sudo apt-get install libncurses5-dev libncursesw5-dev

# **개발 내용 및 결과물**

## **목표**

|  |
| --- |
| **작성요령 (10점)**  **프로젝트의 목표를 기술하세요. 각 단계별 목표를 구체적으로 쓰세요.** |

|  |  |  |
| --- | --- | --- |
| 적용단계 | 내용 | 적용 여부 |
| 1단계 | Map의 구현 | 적용 |
| 2단계 | Snake 표현 및 조작 | 적용 |
| 3단계 | Item 요소의 구현 | 적용 |
| 4단계 | Gate 요소의 구현 | 적용 |
| 5단계 | 점수 요소의 구현 | 적용 |

**프로젝트의 각 적용 단계별 구현 목표를 명확하게 제시한다. 제시한 권고안의 내용을 포함하여, 변경된 부분 등을 구체적으로 단계별 구현 목표를 작성해야 한다.**

과제에서 요구되는 1단계 ~ 5단계 요소들을 차례로 구현하고, 추가 단계까지 구현하여 원활하게 동작하는 Snake game을 개발하고자 한다.

## **1단계 - Map의 구현**

게임 맵을 생성하고 경계를 설정하며 맵 내에 아이템과 게이트를 배치할 수 있도록 함.

## **2단계 - Snake 표현 및 조작**

뱀의 초기 위치와 크기를 설정하고 사용자 입력(방향 키)에 따라 뱀을 이동시킴. 뱀이 자신의 몸이나 맵 경계와 충돌하는지 검사함.

## **3단계 - Item 요소의 구현**

일정 시간마다 맵에 아이템을 생성함. 뱀이 아이템을 먹으면 점수가 증가하고 뱀의 길이가 늘어남.

## **4단계 - Gate 요소의 구현**

맵의 특정 위치에 게이트를 생성함. 뱀이 게이트를 통과하면 다른 위치로 순간 이동함.

## **5단계 - 점수 요소의 구현**

게임 진행 중 현재 점수를 화면에 표시하고 게임 종료 시 최종 점수를 표시함.

## **개발 내용 및 결과물**

### **개발 내용**

|  |
| --- |
| **작성요령 (10점)**  **프로젝트의 수행의 내용을 구체적으로 기술한다. 세부 목표별로 어떤 결과를 어떤 방법으로 달성하였는지를 자세히 기술한다.** |

1. **Map 구현**

**MapManager 클래스**: 이 클래스는 맵의 생성, 업데이트, 표시 기능을 담당함. 맵의 경계를 설정하고 아이템과 게이트를 배치하는 기능을 구현함.

1. **Snake 표현 및 조작**

**Snake 클래스**: 뱀의 초기 위치와 크기를 설정하고 사용자 입력에 따라 뱀을 이동시키는 기능을 구현함. 뱀이 자신의 몸이나 맵의 경계와 충돌하는지를 검사함.

1. **Item 요소 구현**

**Item 클래스**: 일정 시간마다 맵에 아이템을 생성하고 뱀이 아이템을 먹으면 점수를 증가시키고 뱀의 길이를 늘리는 기능을 구현함.

1. **Gate 요소 구현**

**Gate 클래스**: 맵의 특정 위치에 게이트를 생성하고 뱀이 게이트를 통과하면 다른 위치로 순간 이동하게 하는 기능을 구현함.

1. **점수 요소 구현**

**Score 클래스**: 게임 진행 중 현재 점수를 표시하고 게임 종료 시 최종 점수를 표시하는 기능을 구현함.

### **시스템 구조 및 설계도**

|  |
| --- |
| **작성요령 (30점)**  **프로젝트의 각 세부 목표의 주요 기능(알고리즘 등)에 대해서 기술한다. 세부 목표별로 수정한 프로그램 소스 파일을 나열하고, 해당 파일에서 세부 목표를 달성하기 위해 작성한 클래스/함수에 대해 나열하고, 각 요소에 대해 간략한 설명을 작성한다. 또한 각 요소의 개발자를 명시한다.** |

시스템은 주요 클래스들로 구성되어 있으며, 각 클래스의 역할은 다음과 같음:

1. **MapManager 클래스** (MapManager.cpp, MapManager.h)

**기능**: 맵의 생성, 업데이트, 표시

**클래스/함수**: createMap(), updateMap(), displayMap()

**개발자**: 박진오

1. **Snake 클래스** (snake.cpp, Snake.h)

**기능**: 뱀의 이동, 성장, 충돌 검사

**클래스/함수**: moveSnake(), growSnake(), checkCollision()

**개발자**: 김찬중

1. **Item 클래스** (Item.cpp, Item.h)

**기능**: 아이템 생성 및 상호작용

**클래스/함수**: generateItem(), checkItemCollision()

**개발자**: 조성민

1. **Gate 클래스** (Gate.cpp, Gate.h)

**기능**: 게이트 생성 및 이동

**클래스/함수**: createGate(), teleportSnake()

**개발자**: 조성민

1. **Score 클래스** (Score.cpp, Score.h)

**기능**: 점수 계산 및 표시

**클래스/함수**: updateScore(), displayScore()

**개발자**: 박진오

### **활용/개발된 기술**

|  |
| --- |
| **작성요령 (10점)**  **프로젝트 수행에 사용한 외부 기술/라이브러리를 나열하여 작성한다. 각각 기술을 이 프로젝트에 적용할 때, 도움 받거나 해결하고자 하는 기능에 대해 상세히 설명한다.**  **NCURSES / STL 라이브러리 등을 포함하여 설명한다.**  **또한, 이 프로젝트를 수행하면서, 새롭게 고안한 알고리즘 등이 있다면 설명한다.** |

**Ncurses**

**역할 :** 텍스트 기반의 사용자 인터페이스를 제공하는 라이브러리이다. 이 프로젝트에서는 게임 화면을 그리거나, 사용자의 입력을 처리하는 데 사용되었다.

**적용한 기능 :**

게임 보드 그리기 - ncurses 라이브러리를 사용하여 게임 보드를 터미널에 그린다.

사용자 입력 처리 - 키보드 입력을 비동기적으로 처리하여 뱀의 이동을 제어한다.

화면 업데이트 - 게임 상태가 변경될 때마다 화면을 즉시 갱신하여 실시간으로 게임 진행 상황을 보여준다.

**STL (Standard Template Library)**

**역할 :** C++ 표준 라이브러리로, 데이터 구조와 알고리즘을 제공한다. 이 프로젝트에서는 주로 vector, deque, map 등의 컨테이너를 사용하였다.

**적용한 기능 :**

데이터 저장 - vector와 deque를 사용하여 뱀의 몸체 좌표와 같은 동적 데이터 구조를 관리한다.

효율적 접근 및 수정 - map을 사용하여 게임 보드의 상태를 효율적으로 관리한다.

**새롭게 고안한 알고리즘**

**뱀의 이동 알고리즘**

뱀이 이동할 때마다 새로운 머리 좌표를 계산하고, 꼬리 좌표를 제거하는 방식으로 구현하였다. 이 과정에서 충돌감지를 동시에 수행하여 게임 오버 조건을 체크한다.

꼬리를 제거하는 대신, 먹이를 먹었을 때는 꼬리를 제거하지 않음으로써 뱀이 길어지도록 하였다.

**먹이 생성 알고리즘**

게임 보드의 빈 공간을 찾아서 무작위로 먹이를 생성한다. 이때, map을 사용하여 현재 뱀이 차지하고 있는 좌표를 제외하고 먹이가 생성되도록 한다.

### **현실적 제한 요소 및 그 해결 방안**

|  |
| --- |
| **작성요령 (5점)**  **제안된 프로젝트의 단계 별 수행에 있어, 제한 요소를 찾아 작성한다. 해당 제한 요소를 해결하기 위해서 어떤 방법으로 해결하였는지 작성한다.** |

**제한 요소**

**1 . 터미널 환경에서의 그래픽 제한**

설명: 터미널 환경에서는 고해상도의 그래픽을 사용할 수 없고, 단순한 텍스트와 문자 기반의 그래픽만을 사용할 수 있다. 이는 그래픽적으로 풍부한 게임을 만들기 어렵게 한다.

**2 . NCURSES 라이브러리의 제한**

설명: NCURSES는 텍스트 기반의 사용자 인터페이스를 제공하는 라이브러리로, 복잡한 그래픽을 구현하는 데 한계가 있다. 또한, 색상 지원이나 애니메이션 효과 등 현대적인 그래픽 요소를 구현하기에 적합하지 않다.

**해결 방안**

**1 . 텍스트 기반 그래픽의 최적화**

**해결 방법** : NCURSES의 기능을 최대한 활용하여 텍스트 기반의 그래픽을 구현하였다. 예를 들어, 다양한 문자 조합을 사용하여 게임 화면을 구성하고, 텍스트 캐릭터를 이용해 뱀, 아이템, 게이트 등의 게임 요소를 표현하였다.

**구체적 조치**:

**게임 보드 디자인** - 문자를 활용하여 게임 보드를 시각적으로 구분하고, 경계선, 뱀, 아이템 등을 명확하게 표시하였다.

**움직임 및 애니메이션** - 뱀의 이동, 아이템 생성, 게이트 통과 등의 애니메이션 효과를 텍스트 기반으로 구현하여 게임의 동적인 요소를 표현하였다.

**2 . 복잡한 그래픽 요소의 단순화**

**해결 방법:** 복잡한 그래픽 요소를 단순화하여 처리함으로써 터미널 환경의 제약을 극복하였다. 예를 들어, 아이템이나 게이트와 같은 요소를 단순한 문자로 표현하고, 텍스트의 위치를 변경하는 방식으로 다양한 효과를 구현하였다.

**구체적 조치:**

**아이템 및 게이트 -** 아이템을 'G' 또는 'P' 문자로, 게이트를 'O' 문자로 표시하여 시각적으로 구분해 가시성을 높이고 각 요소의 위치를 랜덤으로 생성했다.

**점수 표시 -** 점수를 화면의 특정 위치에 실시간으로 업데이트하여 게임 진행 상황을 쉽게 파악할 수 있도록 하였다.

**3 . 효율적인 입력 처리 및 화면 갱신**

**해결 방법 :** NCURSES의 비동기 입력 처리 기능을 활용하여 사용자의 키보드 입력을 실시간으로 반영하고, 화면을 효율적으로 갱신하였다.

**구체적 조치:**

**비동기 입력 처리 -** getch() 함수를 논블로킹 모드로 설정하여 사용자의 키 입력을 실시간으로 받아들이고, 뱀의 움직임을 즉각적으로 반영하였다.

**화면 갱신 최적화 -** 화면 갱신 주기를 최적화하여 과도한 리프레시를 방지하고, 필요한 부분만 갱신함으로써 성능을 향상시켰다.

**4 메모리 및 성능 최적화**

**해결 방법 :** STL 컨테이너와 효율적인 알고리즘을 사용하여 메모리 사용을 최적화하고, 게임의 성능을 개선하였다.

**구체적 조치:**

**효율적인 데이터 구조 사용 -** 뱀의 몸체 좌표를 저장하는 데 deque를 사용하여 삽입 및 삭제 연산을 효율적으로 처리하였다.

**충돌 검사 최적화 -** 맵의 상태를 효율적으로 관리하기 위해 map을 사용하여 빠른 충돌 검사를 구현하였다.

이와 같은 해결 방안을 통해 터미널 환경과 NCURSES 라이브러리의 제한을 극복하고, 원활하게 동작하는 Snake Game을 개발할 수 있었다. 각 제한 요소를 구체적으로 분석하고, 이를 해결하기 위한 다양한 기술적 접근을 통해 프로젝트를 성공적으로 완수할 수 있었다.

### **결과물 목록**

|  |
| --- |
| **작성요령 (5점)**  **결과물 목록을 작성한다. 목록은 제출하는 파일과 각 파일의 역할을 간략히 설명한다.** |

**1. main.cpp**

**역할 :** 프로젝트의 진입점으로, 게임의 초기화, 메인 루프, 종료 처리 등을 담당하는 파일이다.

**주요 기능:**

**게임 초기화 -** ncurses 라이브러리를 초기화하고, 게임에 필요한 초기 설정을 수행한다.

**메인 게임 루프 -** 게임의 주요 루프를 실행하여 뱀의 이동, 아이템 생성, 충돌 검사 등을 반복적으로 처리한다.

**종료 처리 -** 게임이 종료될 때 ncurses 라이브러리를 종료하고, 필요한 정리 작업을 수행한다.

**특징:** 게임의 전반적인 흐름을 관리하며, 사용자 입력을 받아 뱀을 제어하는 핵심 로직이 포함되어 있다.

**2. Makefile**

**역할:** 프로젝트의 빌드 및 실행을 자동화하는 파일이다.

**주요 기능:**

**컴파일 명령어 정의 -** 프로젝트에 필요한 소스 파일들을 컴파일하여 실행 파일을 생성하는 명령어를 정의한다.

**클린업 명령어 정의 -** 빌드 과정에서 생성된 임시 파일이나 실행 파일을 삭제하는 명령어를 포함한다.

**특징:** 개발자가 간편하게 프로젝트를 빌드하고 실행할 수 있도록 돕는 중요한 도구이다. make 명령어를 통해 손쉽게 프로젝트를 관리할 수 있다.

**3. MapManager.cpp & MapManager.h**

**역할:** 게임 맵의 상태를 관리하고, 맵 관련 기능을 제공하는 클래스가 정의된 파일들이다.

**주요 기능:**

**맵 생성 -** 게임 맵을 초기화하고, 경계를 설정하며, 아이템과 게이트를 배치하는 기능을 제공한다.

**맵 업데이트 -** 게임 진행 중 맵의 상태를 업데이트하여 아이템의 위치나 게이트의 상태를 변경한다.

**맵 표시 -** ncurses를 사용하여 현재 맵의 상태를 터미널에 출력한다.

**특징:** 맵의 구조와 상태를 관리하는 핵심 클래스이며, 게임의 주요 요소들이 맵에 배치되고, 갱신되는 로직을 포함한다.

**4. snake.cpp & Snake.h**

**역할:** 뱀의 상태와 동작을 관리하는 클래스를 정의한 파일들이다.

**주요 기능:**

**뱀의 초기화 -** 뱀의 초기 위치와 크기를 설정한다.

**뱀의 이동 -** 사용자 입력에 따라 뱀의 머리를 새로운 위치로 이동시키고, 꼬리를 따라가게 한다.

**충돌 검사 -** 뱀이 자신의 몸이나 맵의 경계와 충돌하는지 검사하여 게임 오버 조건을 확인한다.

**아이템 먹기:** 뱀이 아이템을 먹으면 뱀의 길이를 늘리거나 줄이고, 점수를 증가시키는 기능을 구현한다.

**특징:** 게임의 주요 캐릭터인 뱀의 모든 동작과 상태를 관리하며, 게임의 핵심 로직이 포함되어 있다.

이와 같은 파일들로 구성된 프로젝트 결과물은 각각의 역할을 명확히 하여, 효율적인 개발과 유지보수를 가능하게 했다. 각 파일은 게임의 특정 기능을 담당하며, 서로 긴밀하게 연동되어 완전한 게임 시스템을 구성했다.

# **자기평가**

|  |
| --- |
| **작성요령 (5점)**  **프로젝트를 수행한 자기 평가를 서술한다. 팀원 개개인의 자기 평가가 포함되어야 하며, 본인의 역할, 프로젝트 수행 시 어려운 점, 도움이 되었던 점, 이 프로젝트 운영에 개선이 필요하다고 생각하는 점을 충분히 서술한다.** |

|  |  |
| --- | --- |
| 박진오 | 이번 프로젝트의 조장을 맡았다.  전체적인 프로젝트 관리와 코드 작성에 기여했고 프로젝트의 방향을 설정하고 팀원 간의 협력을 조율했다. 프로젝트를 진행하면서 팀원들과의 협업과 코드 통합 과정에서 많은 것을 배울 수 있었지 않았을까... 생각이 된다. |
| 김찬중 | 맵과 뱀의 동작을 구현하는 데 주로 기여하였습니다. ncurses 라이브러리를 처음 사용해보아 어려움을 겪었지만, 다양한 튜토리얼과 문서를 통해 문제를 해결해가며 많은 것을 학습했습니다. 특히 뱀의 이동과 충돌 검사 기능 구현에 집중하였습니다. |
| 조성민 | 아이템과 게이트 요소 구현을 맡았습니다. 게임의 재미 요소를 추가하기 위해 다양한 아이템과 게이트의 동작을 설계하고 구현하려고 노력했습니다. 프로젝트를 진행하면서 시간이 부족하고 굉장히 막막했지만 팀장님이 열심히 도와 주셔서 덕분에 포기하지 않고 끝까지 왔습니다. 새로운 알고리즘을 적용해보고 문제를 해결하는 과정에서 짧은 기간에 많은 경험을 쌓았습니다. |

# **참고 문헌**

**참고한 서적, 기사, 기술 문서, 웹페이지를 나열한다.:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **번호** | **종류** | **제목** | **출처** | **발행년도** | **저자** | **기타** |
| 01 | 웹 페이지 | Curses (ncurses) tutorial | https://github.com/mcdaniel/curses\_tutorial | 2024-01 | Patick McDaniel |  |
| 02 | 웹 페이지 | NCURSES Programming HOWTO | https://tldp.org/HOWTO/NCURSES-Programming-HOWTO/ | 2005-06 | ppadala |  |
| 03 | 웹 페이지 | Makefile Tutorial | https://makefiletutorial.com/ |  | Chase Lambert | 저자 깃허브 링크: https://github.com/theicfire |
| 04 | 웹 페이지 | GNU ncurse | https://invisible-island.net/ncurses/announce.html |  | Thomas E. Dickey |  |

# **부록**

|  |
| --- |
| **작성요령 (15점)**  **프로젝트의 결과물을 사용하기 위한 방법에 대해서 작성하세요.** |

## **사용자 매뉴얼**

**프로젝트 실행 :**

1. 터미널을 열고 프로젝트 디렉토리로 이동한다.
2. Makefile을 이용해 컴파일을 진행한다.(make run명령어)
3. ./game 명령어를 입력하여 게임을 실행한다.

**게임 조작 방법 :**

방향키를 사용하여 뱀을 이동시킨다. G(Growth) 아이템을 먹으면 점수가 증가하고 snake 의 진행방향으로 길이가 늘어난다. P(Poison) 아이템을 먹으면 snake 꼬리부분의 길이가 감소한다.

맵에 랜덤으로 생성되는 게이트를 통과하는 것을 포함하여 Growth아이템 먹기, Poison아이템 먹기 등 여러 미션을 통과하면 해당 스테이지가 클리어되고 다음 스테이지로 넘어간다. 3스테이지까지 완료하면 게임이 종료된다.

## **설치 방법**

**1. Ubuntu에서 ncurses 설치**

**패키지 목록 업데이트**

최신 패키지 목록을 가져오기 위해 터미널에 아래 명령어를 입력합니다.

sudo apt-get update

**ncurses 라이브러리 설치**

ncurses 라이브러리를 설치하기 위해 아래 명령어를 입력합니다.

sudo apt-get install libncurses5-dev libncursesw5-dev

**2. 프로그램 빌드 및 실행**

**프로젝트 디렉토리로 이동**

터미널에서 cd 명령어를 사용하여 프로젝트 디렉토리로 이동합니다.

팀장 노트북 기준 : cd /mnt/c/Users/spsps/Documents/snake/game

**Makefile을 이용한 컴파일**

프로젝트 디렉토리로 이동한 후, Makefile을 사용하여 프로젝트를 컴파일합니다. ‘make’ 명령어를 입력하여 컴파일을 시작한다.

컴파일 과정에서 오류가 발생하지 않는지 확인한다. 성공적으로 컴파일되면 ./game 실행 파일이 생성된다.

**게임 실행**

컴파일이 완료되면 ./game 명령어를 입력하여 게임을 실행합니다.

게임이 시작되면 터미널 창에 게임 화면이 나타나고, 게임을 플레이할 수 있습니다.