**C++프로그래밍**

**프로젝트**

|  |  |
| --- | --- |
| 프로젝트 명 | *Snake Game* |
| 팀 명 | *1분반 1팀* |
| 문서 제목 | 최종 |

|  |  |
| --- | --- |
| **Version** | 1.4 |
| **Date** | 2020-JunE-21 |

|  |  |
| --- | --- |
| **팀원** | 김수연 (팀장) |
| 권소예 |
| 김성훈 |
| 박세호 |
|  |

|  |
| --- |
| **CONFIDENTIALITY/SECURITY WARNING**  이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 C++프로그래밍 수강 학생 중 프로젝트 “Snake Game”을 수행하는 팀 “1분반 1팀”의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 “1분반 1팀”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다. |

**문서 정보 / 수정 내역**

|  |  |
| --- | --- |
| **Filename** | 1분반1팀-프로젝트수행결과보고서.docx |
| **원안작성자** | 김수연 |
| **수정작업자** | 김수연, 박세호, 김성훈 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 수정날짜 | 대표수정자 | Revision | 추가/수정 항목 | 내 용 |
| 2020-06-10 | 김수연 | 1.0 | 최초 작성 | 2.1 목표 작성 |
| 2020-06-17 | 김수연 | 1.1 | 내용 추가 | Stage2 개발 내용 작성 |
| 2020-06-20 | 김성훈 | 1.2 | 내용 추가 | - mission board 이동 및 크기 가변적으로 조정  - score board 이동 및 크기 가변적으로 조정  - 맵 위치 이동 및 크기 가변적으로 조정  - 2, 3, 4 단계 추가  - 메시지 위치 조정  - Makefile 이 Mac OS 에서도 동작하도록 수정  - identifier 명확하게 수정 |
| 2020-6-21 | 박세호 | 1.3 | 내용추가 | Stage4개발 내용 작성 |
| 2020-06-21 | 김성훈 | 1.4 | 내용 추가 | - mission board 에 현황 업데이트 되도록 구현  - 미션 완료 시 맵 변경되도록 구현  - 각각의 맵에 장애물을 맵 중간에 추가  - 화면이 항상 clear and draw 되도록 구현 |

**목 차**

[1 개요 5](#_Toc43673457)

[1.1 소스 파일 구조 5](#_Toc43673458)

[1.2 실행 파일 구조 5](#_Toc43673459)

[1.3 개발 및 실행 환경 5](#_Toc43673460)

[1.4 사용하고 있는 외부 라이브러리 5](#_Toc43673461)

[1.5 획득/설치 방법 6](#_Toc43673462)

[2 개발 내용 및 결과물 7](#_Toc43673463)

[2.1 목표 7](#_Toc43673464)

[2.2 개발 내용 및 결과물 9](#_Toc43673465)

[2.2.1 개발 내용 9](#_Toc43673466)

[2.2.2 시스템 구조 및 설계도 9](#_Toc43673467)

[2.2.3 활용/개발된 기술 12](#_Toc43673468)

[2.2.4 현실적 제한 요소 및 그 해결 방안 13](#_Toc43673469)

[2.2.5 결과물 목록 14](#_Toc43673470)

[3 자기평가 15](#_Toc43673471)

[4 참고 문헌 15](#_Toc43673472)

[5 부록 16](#_Toc43673473)

[5.1 사용자 매뉴얼 16](#_Toc43673474)

[5.2 설치 방법 17](#_Toc43673475)

# 개요

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

## 소스 파일 구조

* Makefile
* main.cpp

게임 초기화 및 게임 루프 실행

* main.h

공통 헤더 파일

* snake.cpp, snake.h

맵 데이터로부터 위치를 초기화한 Snake 객체 생성, 매 틱마다 실행되는 tick 함수 정의, snake를 화면에 출력하는 Snake 클래스

* map.cpp, map.h

맵 데이터로부터 초기화 및 맵 크기 정보, 맵 출력 기능을 제공하는 클래스

* board.cpp

score board, mission board 를 가변적으로 그려주기 위한 함수를 정의

## 실행 파일 구조

* main
* main.o
* map.o
* snake.o
* board.o
* ncurses 라이브러리(우분투에서는 libncurses.a

## 개발 및 실행 환경

* 운영체제 : 우분투 16.04, Mac OS

## 1.4 사용하고 있는 외부 라이브러리

* ncursesw

유니코드 출력을 위한 ncurses의 유니코드 버전

## 1.5 획득/설치 방법

다음과 같은 명령을 사용하여 라이브러리를 설치한다.

sudo apt-get install libncurses5-dev libncursesw5-dev

cd src && make && ./main

# 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/미적용 |
| 4단계 | Gate 요소의 구현 | 적용 |
| 5단계 | 점수 요소의 구현 | 적용 |

1단계

* NCurses Library 함수들을 사용하여 2차원 배열로 된 Snake Map을 Game 화면으로 표시하는 프로그램을 완성한다.
  + 2차원 배열 내에 0, 1, 2, 3, 4의 정수로 각각 바탕, Wall, Immune Wall, Head, Body를 표현한다.

2단계

* 1단계의 맵 위에 Snake를 표시하고, 화살표를 입력 받아 Snake가 움직이도록 프로그램을 완성한다.
* 다음과 같은 Game Rule #1을 만족한다.
  + Snake는 진행방향의 반대방향(Tail 방향)으로 이동할 수 없고, Tail 방향으로 이동 시 실패(Game over)한다.
  + Head 방향 이동은 무시하고, Tail 방향으로 이동할 수 없다.
  + Snake는 자신의 Body를 통과할 수 없고, 벽(Wall)을 통과할 수 없다.
  + Head 방향 이동은 일정시간(틱) 0.5초에 의해 이동한다.

3단계

* 2단계 프로그램에서, Map 위에 Growth Item와 Poison Item을 출현하도록 수정한다.
* 다음과 같은 Game Rule #2를 만족한다.
  + Snake의 이동 방향에 Item이 놓여 있는 경우, Snake가 Item을 획득한다.
    - Growth Item의 경우 몸의 길이(Tail)가 1 증가한다.
    - Poison Item의 경우 몸의 길이(Tail)가 1 감소한다. 몸의 길이가 3보다 작아지면 실패한다.
    - Growth Item과 Poison Item은 Snake Body가 있지 않은 임의의 위치에 출현한다.
    - 출현 후 일정시간(틱)이 지나면 사라지고 다른 위치에 나타나야 한다.
    - 동시에 출현할 수 있는 Item의 수는 3개로 제한한다.
* Growth Item과 Poison Item을 Map 배열에 표현할 때 값을 정한다.
* 화면 상에 표현 시, 색을 달리하여 구분할 수 있도록 한다.
* Map Data에서 Growth Item은 5, Poison Item은 6과 같이 구분할 수 있도록 한다.

4단계

* 3단계 프로그램에서, Map의 Wall의 임의의 위치에 한 쌍의 Gate가 출현할 수 있도록 변경하고, 각 Gate에 Snake가 통과할 수 있도록 수정한다.
* 다음과 같은 Game Rule #3을 만족한다.
  + Gate는 두 개가 한 쌍이며 서로 겹치지 않는다. 한 번에 한 쌍만 나타난다.
  + Gate는 임의의 위치에 있는 벽에서 나타난다.
  + Gate에 Snake가 진입하면, 다른 Gate로 진출한다.
  + Gate에 Snake가 진입 중인 경우, Gate는 사라지지 않고, 다른 위치에서 Gate가 나타나지 않는다.
* 다음과 같은 Game Rule #4를 만족한다.
  + Gate가 나타나는 벽이 가장자리에 있을 때
    - 항상 Map의 안쪽 방향으로 진출한다. (고정규칙)
    - 상단 벽 -> 아래 방향
    - 하단 벽 -> 위 방향
    - 좌측 벽 -> 오른쪽 방향
    - 우측 벽 -> 왼쪽 방향
  + Gate가 나타나는 벽이 Map의 가운데 있을 때 다음의 순서로 진출
    - 진입 방향과 일치하는 방향이 우선
    - 진입 방향의 시계방향으로 회전하는 방향
    - 진입 방향의 역시계방향으로 회전하는 방향
    - 진입 방향과 반대방향
* 다음과 같은 Game Rule #5를 만족한다.
  + Wall과 Immune Wall, Gate의 출현 방법
    - Wall은 Gate로 변할 수 있고, Immune Wall은 Gate로 변할 수 없다.
    - 모든 Wall은 Snake가 통과할 수 없으며, Snake Head와 충돌 시 실패한다.
    - Gate의 출현은 몸의 최대 길이 10을 초과하면 발생한다.

5단계

* 4단계 프로그램에서, 우측에 게임 점수를 표시하는 화면을 구성한다.
* 다음과 같은 Game Rule #6을 만족한다.
  + 다음과 같은 점수를 계산한다.
    - 게임 중 몸의 최대길이 계산
    - 게임 중 획득한 Growth Item의 수
    - 게임 중 획득한 Poison Item의 수
    - 게임 중 Gate 사용 횟수
    - 게임 시간(Seconds로 계산)
  + 게임 방법
    - 주어진 미션 달성하기
  + 미션
    - 각 점수 항목별로 목표치 도달 시 게임 종료
* 미션
  + 구성한 Map의 정의에 고정 값을 주거나, 매 게임마다 임의의 값을 주는 방식으로 처리한다.
* 미션을 달성하면 다음 Map으로 진행하도록 프로그램을 완성한다.
  + Stage는 최소 4개로 구성하고, 각 Stage의 Map은 서로 달라야 한다.
* 추가옵션
  + 배경음 및 효과음 삽입하기
    - 배경음악은 게임 시작부터 시작하여 게임 종료까지 계속 재생
    - 효과음의 종류
      * 게임 시작 시
      * Item(Growth Item, Poison Item)을 획득 시
      * Fail(게임 오버)
      * Gate 들어갈 때와 나갈 때
      * Mission 달성 시
      * 다음 Stage로 넘어갈 시
      * 모든 Stage 완료 시

## 개발 내용 및 결과물

### 개발 내용

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

1. **1단계 개발 내용**

* 맵을 int의 배열의 배열로 표현했다.
* 변과 모서리, snake를 나타내기 위한 적절한 유니코드 문자를 찾아내어 적용하였다.

유니코드 문자 출력 시 윈도우의 가로 두 칸을 차지하므로 이에 맞게 출력 루틴을 작성하였다.

* 각 컬러 팔레트를 서로 다른 색상으로 초기화하였다.
* 기본적인 score board와 mission board를 고정 배치하였다.

1. **2단계 개발 내용**

* 단일 프로그램 파일을 여러 클래스(Map, Snake, Failure)로 분할 설계하여 작성하였다.
* 소스 코드를 cpp 파일과 헤더 파일로 분리하였다.
* 맵의 크기를 유동적으로 하기 위해 맵의 표현을 위해 vector<vector<int>> 타입을 사용하였다.
* Game Loop와 tick 멤버 함수를 추가하여 snake의 이동을 처리했다.
* Snake 객체를 생성할 때 mapData 정보를 받아들여 Snake의 위치 정보를 좌표들의 deque로 표현하였다. deque를 사용하면 snake의 이동을 간편하고 효율적으로 처리할 수 있기 때문이다. Snake의 Head와 바로 이어진 Body의 상대적인 위치를 통해 Snake의 초기 이동 방향을 검출하였다.
* enum Direction을 도입하여 다.
* 논리적인 컬러 팔레트 이름을 도입하여 enum Pallete 타입으로 정의하고 이를 한 군데에 정의함으로써 한 군데에서만 색상을 변경하면 전체에 적용되도록 하고 가독성을 높였다.
* Makefile을 도입하였다.
* snake의 Head가 다음 틱에 있을 좌표를 계산하여 wall conflict와 body conflict를 검출하였다.

1. **3단계 개발 내용**
2. **4단계 개발 내용**

* gate를 랜덤으로 생성하기 위해 rand()를 사용했고 이를 map 배열에 지정해 gate를 생성시켜줬다.
* gate의 좌표 두 개를 return 해주기 위해서 구조체를 만들어 gate의 x,y를 관리 하였다.
* tick 마다 snake가 gate를 통과하는지 감시해 gate가 snake가 통과하는 도중에는 바뀌지 않게 해주었다.
* 처음에 10tick이 지나면 gate가 생성되게 만들었다.
* gate가 snake를 통과하면 다음 gate를 랜덤으로 지정해주었다.

1. **5단계 개발 내용**

* 5단계의 첫 번째 핵심 목표는 맵과 점수판, 그리고 미션판이 조화롭게 잘 보이는지였고, 두 번째는 미션판의 목표가 모두 달성되면 다음 맵으로 이동하는 것이었다.
* 조화롭게 그려지기 위해서는 맵의 크기에 비례해서 점수판, 미션판을 위한 멤버 변수들이 필요했고, 목표가 달성되는 것을 확인하기 위해서는 특정 액션이 일어날 때마다 체크가 필요했으며 이를 위한 정보들은 Snake 객체에 멤버 변수로 저장돼야 했다.
* 이 기능들은 모두 복잡하지 않았기 때문에 어려운 것은 없었다. 하지만 미션이 완료되면 맵이 변경되며 점수판, 미션판의 위치, 크기 모두 변경하는 과정에서 문제가 발생했으며 윈도우들의 그림이 충돌되는 문제가 있었다. 이는 ncurses에 그려진 것을 모두 초기화하는 clear 함수를 통해 해결했고 모두 다시 그려주는 과정을 통해 해결됐다.

### 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.** |

1. **1단계 목표**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파일** | **클래스** | **함수** | **설명** | **개발자** |
| **snake.cpp** |  |  | **단일 프로그램 파일** | **김수연** |
| **main()** | **게임 초기화 및 초기 맵 출력** | **김수연** |

1. **2단계 목표**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파일** | **클래스** | **함수** | **설명** | **개발자** |
| **main.cpp** |  |  |  | **김수연** |
| **main()** | **게임 초기화 및 게임 루프 실행** | **김수연** |
| **initGame()** |  | **김수연** |
| **main.h** |  |  | **공통 헤더 파일** | **김수연** |
| **Failure** | **Failure()** | **snake 게임 오버를 표현하는 클래스** | **김수연** |
| **enum Pallete** |  | **컬러 팔레트 열거형 상수** | **김수연** |
| **snake.cpp, snake.h** | **Snake** |  | **Snake 객체의 표현과 동작** | **김수연** |
| **Snake()** | **맵 데이터로부터 위치와 이동 방향을 초기화한 Snake 객체 생성** | **김수연** |
| **findBody()** | **mapData로부터 snake의 body를 재귀적으로 찾아내기** | **김수연** |
| **tick()** | **한 tick 내에 수행하는 snake의 동작 정의** | **김수연** |
| **display()** | **Snake 객체를 화면에 출력** | **김수연** |
| **checkWallConflict()** | **Snake 객체의 head가 Wall에 충돌하였는지 검사** | **김수연** |
| **checkBodyConflict()** | **Snake 객체의 head가 자신의 body에 충돌하였는지 검사** | **김수연** |
| **map.cpp, map.h** | **Map** | **setbgMap()** | **mapData로 현재 배경 맵 설정** | **김수연** |
| **getRowSize()** | **mapData의 column size 반환** | **김수연** |
| **getColSize()** | **mapData의 column size 반환** | **김수연** |
| **operator[]** | **Map 객체에서 직접 현재 맵의 인덱스 번째의 한 행에 대한 레퍼런스를 반환** | **김수연** |
| **display()** | **맵을 화면에 출력** | **김수연** |

1. **3단계 목표**
2. **4단계 목표**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **파일** | **클래스** | **함수** | **설명** | **개발자** |
| **snake.cpp, snake.h** | **Snake** |  | **Snake 객체의 표현과 동작** | **김수연**  **,박세호** |
| **Struct \_Gate** | **Gate의 좌표를 저장하는 구조체** | **박세호** |
| **checkGatein()** | **snake가 Gate를 통과 여부와 통과했다면 반대 게이트의 좌표 return** | **박세호** |
| **getGate1(), getGate2()**  **setGate1(), setGate2()** | **두 게이트 정보 get, set하는 함수** | **박세호** |
| **snakelength()** | **snake의 길이 return** | **박세호** |
| **RandomGate()** | **Gate를 랜덤으로 생성하는 함수** | **박세호** |
| **getGatepass(),**  **setGaetPass()** | **gatePass(snake의 gate통과여부) set, get하는 함수** | **박세호** |
| **map.cpp, map.h** | **Map** | **setbgMap(),setbgMapraw(),setBgMapgate(),** | **mapData와 로 현재 배경 맵 설정, mapData로 현재 배경맵 초기화, map Data와 gate인자로 gate가 포함된 배경맵 설정** | **김수연,박세호** |
| **display()** | **맵을 화면에 출력** | **김수연,**  **박세호** |

1. **5단계 목표**

김성훈 작업 내용

* board.cpp: draw\_board 를 구현해서 맵 우측에 항상 점수와 미션 관련된 것이 알맞은 크기로 노출되게 함
* map.h: 현재 맵이 몇 번째 인덱스인지 저장하는 멤버 변수 추가
* main.cpp: 반복문에서 항상 window 를 clear 시켜서 다음 단계를 그리거나 점수판을 업데이트 할 때 깨끗하게 출력해줌
* snake.h: 현재 뱀 상태(획득한 아이템 또는 통과한 게이트)를 표현하기 위한 멤버 변수 추가
* snake.cpp: 현재 뱀이 어떠한 맵에 있는지 표현 가능하도록 구현 변경
* map.cpp: 맵이 실시간으로 변경될 수 있도록 구현 및 단계별 맵 구현, 장애물의 위치, 맵의 크기를 변경

### 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: 터미널에서 TUI 를 보여주기 위한 라이브러리이며 API 를 제공해준다. 또한 화면의 변화에 대해 최적화 된 구현으로 인해 성능이 좋다.

### 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.** |

* **유니코드 문자 출력 문제**

**유니코드 문자는 출력될 때 두 칸을 차지하는데, 문자가 반쪽만 출력되는 문제가 있어서 배경을 아스키 문자로 두 번 출력하여 지운 후에 다시 유니코드 문자를 출력하는 방법으로 해결함(Snake::display() 멤버 함수 코드 참고)**

* 색상 부족 문제
* Mac OS 와 Linux 의 library 관련 사용이 다르기 때문에 Makefile 에 조건별로 동작하도록 기능 구현

### 결과물 목록

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

* Makefile
* main.cpp

게임 초기화 및 게임 루프 실행

* main.h

공통 헤더 파일

* snake.cpp, snake.h

맵 데이터로부터 위치를 초기화한 Snake 객체 생성, 매 틱마다 실행되는 tick 함수 정의, snake를 화면에 출력하는 Snake 클래스

* map.cpp, map.h

맵 데이터로부터 초기화 및 맵 크기 정보, 맵 출력 기능을 제공하는 클래스

* board.cpp

score board, mission board 를 가변적으로 그려주기 위한 함수를 정의

* 1분반1팀-프로젝트수행결과보고서.docx

# 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.** |

**박세호: gate를 구현하는 단계를 맡았다.이 때 코드를 난잡하게 해 보기 힘들게 짠 경향이 있는 것 같다.**

**어려웠던 점은 gate가 통과하는 것을 판단해 gate의 연결을 유지하는 부분이었다. 직접 게임을 만들어 보며 코딩능력이 향상된 것 같다.**

김성훈: Mac OS 에서 개발할 때 Linux 와 다른 환경으로 인해 Makefile 을 수정하게 됐는데 익숙하지 않은 작업이라 시간이 많이 소요됐다. 이로 인해 Makefile 작성에 대한 문법들을 매뉴얼을 통해 공부했으며 매뉴얼을 읽으며 개발하는 실력이 향상된 것 같다.

C++ 의 작성은 아직도 익숙하지 않았기 때문에 코드의 가독성이 좋지 않았다. 이는 충분한 문법에 대한 이해가 없었기 때문인데 조금 아쉽다. 부족한 부분은 교수님이 올려주신 강의를 통해 보충할 예정이다.

김수연: 처음에는 snake가 움직이지 않고 가만히 있어서 작업이 잘 진행되지 않을까봐 걱정됐다. 나중에는 snake가 움직이긴 하는데 자꾸만 자신의 자국을 남겨서 이것을 지우는 데에 신경을 써야 했다. snake 게임의 기반이 되는 Map 클래스와 Snake 클래스를 작성하고 다른 팀원들과 협업하면서, 이러한 기반 클래스들과 멤버 함수들을 이용하여 다른 기능을 더 간편하게 구현할 수 있음을 알았다. 그리고 헤더 파일을 별도로 작성함을 통해 프로그램의 전반적인 이해를 확실하게 해준다는 것을 깨달았다. 특히 헤더 파일은 내용이 적어서 주석들이 한눈에 들어오는 장점이 있어서 앞으로도 잘 활용해야겠다. 프로젝트를 진행하면서 답답한 것도 있고 힘든 점도 있었지만 게임이 점점 잘 구축되는 것이 보람찼다.

# 참고 문헌

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 번호 | 종류 | 제목 | 출처 | 발행년도 | 저자 | 기타 |
| 1 | 웹페이지 | sleep\_for | <https://en.cppreference.com/w/cpp/thread/sleep_for> |  |  |  |
|  |  |  |  |  |  |  |

# 부록

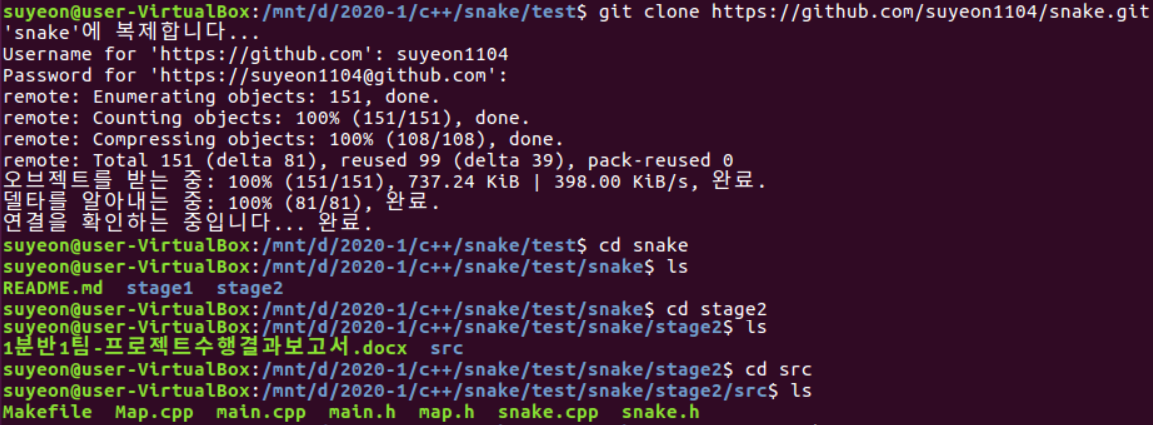
|  |
| --- |
| **작성요령 (15점)**  **프로젝트의 결과물을 사용하기 위한 방법에 대해서 작성하세요.** |

## 사용자 매뉴얼

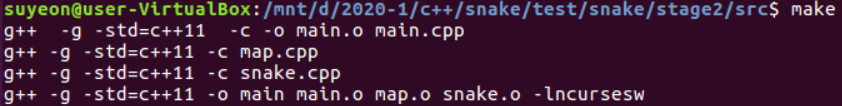
1. 커서를 이용해서 뱀의 위치를 조정할 수 있다.
2. 뱀은 0.3 초마다 이동한다.
3. 게이트는 분홍색으로 표시되며 게이트에 뱀의 머리가 들어가게 되면 다른 게이트를 통해 나온다.
4. 회색 벽에 뱀이 부딪히면 죽는다.
5. 뱀은 현재 진행 방향의 반대편으로 이동하면 죽는다.
6. 우측의 미션에 모두 V 표시가 되면 다음 맵으로 이동한다.

## 설치 방법

1. Github에서 프로젝트 저장소 clone하기



1. Linux 터미널 창에서 해당 폴더에 들어간 후 다음과 같이 make 실행



1. 다음과 같이 게임 프로그램 실행



1. 게임 시작

스크린샷이(가) 표시된 사진

자동 생성된 설명