


C++프로그래밍

프로젝트

프로젝트 명	<i>SnakeGame</i>
팀 명	팀C
문서 제목	결과보고서

Version	1.2
Date	2022-JUN-16

팀원	박 태 범 (팀장)
	양 지 선

 국민대학교 소프트웨어학부 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16


CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 **C++프로그래밍** 수강 학생 중 프로젝트 "**SnakeGame**"를 수행하는 팀 "**팀C**"의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 "**팀C**"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역

Filename	팀C-프로젝트수행결과보고서.doc
원안작성자	양지선
수정작업자	박태범


수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2022-06-05	양지선	1.0	최초 작성	
2022-06-16	박태범	1.1	내용 작성	시스템 구조 및 설계도, 활용된 기술 추가
2022-06-16	박태범	1.2	최종 수정	

	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

국민대학교
소프트웨어학부
C++프로그래밍

목 차

<u>1</u>	<u>개요</u>	4
<u>1.1</u>	<u>프로젝트 개요</u>	4
<u>1.2</u>	<u>사용 라이브러리</u>	4
<u>1.3</u>	<u>라이브러리 설치방법(Ubuntu 기준)</u>	4
<u>2</u>	<u>개발 내용 및 결과물</u>	4
<u>2.1</u>	<u>목표</u>	4
<u>2.2</u>	<u>개발 내용 및 결과물</u>	6
<u>2.2.1</u>	<u>개발 내용</u>	6
<u>2.2.2</u>	<u>시스템 구조 및 설계도</u>	8
<u>2.2.3</u>	<u>활용/개발된 기술</u>	24
<u>2.2.4</u>	<u>현실적 제한 요소 및 그 해결 방안</u>	24
<u>2.2.5</u>	<u>결과물 목록</u>	24
<u>3</u>	<u>자기평가</u>	25
<u>4</u>	<u>참고 문헌</u>	25
<u>5</u>	<u>부록</u>	26
<u>5.1</u>	<u>사용자 매뉴얼</u>	26
<u>5.2</u>	<u>설치 방법</u>	28

 <div> 국민대학교 소프트웨어부 C++프로그래밍 </div>	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

1 개요

평가기준 (10점)

프로젝트를 완성하기 위해 사용한 개발 방법을 기술하세요.

또한 사용하고 있는 외부 라이브러리와 해당 라이브러리를 획득/설치하는 방법을 기술하세요.

1.1 프로젝트 개요

이번 프로젝트 'SnakeGame'은 ubuntu 환경에서 C++을 이용하여 개발되었다. 프로젝트 파일은 snake game을 실행시키는 'main.cpp', snake game의 구현 파일 'snake.cpp', snake.cpp의 헤더파일 'snake.h'로 구성되어있다 .

1.2 사용 라이브러리

- Ncurses

1.3 라이브러리 설치방법(Ubuntu 기준)

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

2 개발 내용 및 결과물

2.1 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

 국민대학교 소프트웨어학부 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

[1단계] Map의 구현

- 4종류의 Map을 생성한다.
- 각 Map은 Wall과 Immune Wall, snake head와 body로 구성되어 있다.

[2단계] Snake 표현 및 조작

- 사용자의 입력에 따라 snake의 진행 방향을 변경한다. 단, 진행방향과 반대 방향의 키를 입력받으면 게임이 종료된다.
- snake head가 Wall이나 snake body와 충돌하면 게임이 종료되도록 한다.
- snake의 길이가 3 미만으로 내려가게 되면 게임을 종료시킨다.

[3단계] Item 요소의 구현


- Growth Item과 Poison Item을 출현시킨다.
- Growth Item과 Poison Item은 특정 시간이 지나면 다른 위치에 새로 생성된다.
- Growth Item을 획득하면 snake tail 길이를 1 증가시킨다.
- Poison Item을 획득하면 snake tail 길이를 1 감소시킨다. 여기서, snake의 길이가 3보다 작아지면 게임을 종료시킨다.

[4단계] Gate 요소의 구현

- Map의 Wall 임의의 위치에 한 쌍(두 개)의 Gate가 출현될 수 있도록 한다.
- Gate는 특정 시간이 지나면 다른 위치에 새로 생성된다.
- Gate에 Snake가 진입하면 다른 Gate로 진출할 수 있도록 한다.
- Gate에 Snake가 진입 중인 경우 Gate는 사라지지 않는다.

[5단계] 점수 요소의 구현

- 각 stage마다 임의의 Mission 값을 설정한다.
- 게임화면 우측 상단에 Score Board, 하단에 Mission창을 구현한다.
- Score Board에는 'B', '+', '-', 'G'의 값이 나타난다.
- Mission창은 'B', '+', '-', 'G'의 목표치를 보여주며 Mission을 달성하면 다음 stage로 넘어간다.

 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

2.2 개발 내용 및 결과물

2.2.1 개발 내용

작성요령 (10점)

프로젝트의 수행의 내용을 구체적으로 기술한다. 세부 목표별로 어떤 결과를 어떤 방법으로 달성하였는지를 자세히 기술한다.

[1단계] Map의 구현

- SnakeGame의 Stage는 1부터 4까지로 Stage마다 서로 다른 구조의 26 X 51크기의 Map 4개를 생성하였다.
- Map은 char type 배열로 '0','1','2','3','4','5','6','7'로 구성하였다. '0'은 공백으로 snake가 지나다닐 수 있는 길이다. '1'은 Wall, '2'는 Immune Wall이다. '3'과 '4'는 각각 snake의 head와 body를 표현하며, '5'는 Growth Item, '6'은 Poison Item, '7'은 Gate를 표현한다.
- Map의 Wall은 '#', Immune Wall은 '*'로 구현하였으며 글자색과 글자배경색은 흰색으로 구현하였다.

[2단계] Snake 표현 및 조작

- 사용자의 입력(←,↑,↓,→)을 받아 snake의 이동 방향을 설정한다. 단, 진행방향과 반대 방향의 키를 입력받으면 게임이 종료된다.
- snake head('3')가 Wall('1')이나 snake body('4')와 충돌하면 게임이 종료되도록 하였다.
- snake의 head는 'O' tail은 'o'으로 표현한다.

[3단계] Item 요소의 구현

- Map의 값이 '0'인 곳에 Growth Item('5')과 Poison Item('6')을 출현시킨다.
- Item마다 Time을 지정해 시간이 지나면 다른 위치로 출현될 수 있도록 한다.
- Growth Item('5')을 획득하면 snake tail('4') 길이를 1 증가시킨다.
- Poison Item('6')을 획득하면 snake tail('4') 길이를 1 감소시킨다. 여기서, snake의 길이가 3보다 작아지면 게임을 종료시킨다.
- Growth Item('5')은 'G', 파란색으로 표현, Poison Item('6')은 'P', 빨간색으로 표현하여 서로 구분될 수 있도록 한다.

[4단계] Gate 요소의 구현

- Map의 Wall 임의의 위치에 한 쌍(두 개)의 Gate가 출현될 수 있도록 한다.
- Gate의 Time을 지정해 시간이 지나면 다른 위치로 출현될 수 있도록 한다.
- Gate에 Snake가 진입하면 다른 Gate로 진출할 수 있도록 한다. 이 때, 진출하는 Gate가 가장자리 Wall에 위치한다면 snake는 Map의 안쪽 방향으로 진출한다. 진출하는 Gate가 가


 국민대학교 소프트웨어학부 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

장자리 Wall에 위치하지 않았다면, 우선적으로 진입 방향과 일치하는 방향으로 진출한다. 그리고 진입 방향의 시계방향으로 회전하는 방향, 역시계방향으로 회전하는 방향, 반대방향 순으로 진출한다.

- Gate에 Snake가 진입 중인 경우 Gate는 사라지면 안되며 다른 위치에 Gate가 생성돼서도 안된다.
- Gate는 '!'로 글자색 흰색, 글자배경색은 노란색으로 표현하여 Wall과 구분될 수 있도록 한다.

[5단계] 점수 요소의 구현

- 각 stage마다 임의의 Mission 값을 설정한다.
- 게임화면 우측 상단에 Score Board, 하단에 Mission창을 구현한다.
- Score Board에는 'B', '+', '-', 'G'의 값이 나타난다. 게임 중 몸의 최대길이를 계산한 'B' 값은 (현재 길이)/(최대 길이)로 나타난다. '+'는 게임 중 획득한 Growth Item의 개수, '-'는 Poison Item의 개수, 'G'는 Gate 사용 횟수를 나타낸다.
- Mission창은 'B', '+', '-', 'G'의 목표치를 보여주며 목표치를 달성하면 해당 미션의 옆에 체크(v) 표시가 된다. 모든 미션을 달성하면 다음 stage로 넘어간다.

 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

2.2.2 시스템 구조 및 설계도

작성요령 (30 점)

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. Snake class / 개발: 박태범


- Snake 클래스에 게임에 필요한 전반적인 기능들을 구현하였다.

```
#include <iostream>
#include <ncurses.h>
#ifndef SNAKE_H
#define SNAKE_H

#define UP 0
#define DOWN 1
#define LEFT 2
#define RIGHT 3
#define SUBMIT 4

class snake{
public:
    bool gameOver;
    int growthCnt, growthItemTime, growthItemX, growthItemY;
    int poisonCnt, poisonItemTime, poisonItemX, poisonItemY;
    int gateX1, gateY1, gateX2, gateY2, gateTime;
    int gItemCnt, pItemCnt, passGateCnt, maxLength, currentLength;
    int nTail, current_tick;
    int goalGItem, goalPItem, goalPassGate, goalMaxLength;
    int _x;
    int _y;
    snake();
    ~snake();
    void start();
    int keyControl();
    int titleDraw();
    void infoDraw();
    int stageSelect();
    void drawMap(int* x, int* y);
    void gLoop(int s);
    void move(int* x, int* y);
    void growthItem();
    void poisonItem();
    void openGate();
    bool missionCheck();
    void victoryDraw();
    int failDraw();
    void scoreBoard();
    int getScore();
    void stageClear();
};

#endif
```


 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

2. Snake 생성자, 소멸자 / 개발: 박태범

- Snake 의 생성자를 통해 기본적인 것들을 설정해준다.

```
snake::snake(){
    initscr();
    keypad(stdscr, TRUE);
    curs_set(0);
    noecho();


    srand(time(0));
    start_color();
    init_pair(1, COLOR_WHITE, COLOR_CYAN);
    init_pair(2, COLOR_WHITE, COLOR_YELLOW);
    init_pair(3, COLOR_MAGENTA, COLOR_CYAN);
    init_pair(4, COLOR_RED, COLOR_CYAN);
    init_pair(5, COLOR_BLUE, COLOR_CYAN);
    init_pair(6, COLOR_BLACK, COLOR_CYAN);
    init_pair(7, COLOR_WHITE, COLOR_WHITE);
}
snake::~snake(){
    endwin();
}
```

3. Key Control / 개발: 박태범

- 사용자의 입력을 받기 위해 방향키마다 특정 값을 return 하도록 설정하였다.

```
int snake::keyControl(){
    int tmp = getch();

    switch (tmp){
        case KEY_LEFT:
            return LEFT;
            break;
        case KEY_UP:
            return UP;
            break;
        case KEY_DOWN:
            return DOWN;
            break;
        case KEY_RIGHT:
            return RIGHT;
            break;
        case ' ':
            return SUBMIT;
            break;
    }
    return 0;
}
```

 국민대학교 소프트웨어학부 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

4. Start / 개발: 박태범

- 게임을 실행시키기 위한 코드이다. titleDraw 함수를 실행하고 return 되는 값에 따라 이후의 작업을 진행한다.
- 최종적으로 Stage 가 선택이 되면 해당 스테이지부터 게임을 시작할 수 있도록 하였다.

```

void snake::start()
{
    int menuCode = titleDraw();
    if (menuCode == 0){
        int stage = stageSelect();
        switch (stage){
            case 0:
                gLoop(0);
                break;
            case 2:
                gLoop(1);
                break;
            case 4:
                gLoop(2);
                break;
            case 6:
                gLoop(3);
                break;
            case 8:
                clear();
                start();
                break;
        }
    }
    else if (menuCode == 2){
        infoDraw();
    }
    else if (menuCode == 4){
        endwin();
    }
}

```

 국민대학교 소프트웨어학부 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

4.1 Title Draw / 개발: 박태범

- titleDraw 함수가 호출되면 메인화면을 출력하고 좌표에 따라 메뉴를 선택할 수 있게끔 제작하였다.

```
// 초기 타이틀 화면
int snake::titleDraw(){
    printf("\n\n\n\n");
    printf("      ###      #      #      #      #      #      #      \n");
    printf("      #      # #      # #      # #      #      \n");
    printf("      ####      # #      #####      #      \n");
    printf("      #      # #      # #      # #      #      \n");
    printf("      ####      #      #      #      #      #      \n");
    bkgd(COLOR_PAIR(1));

    int y = 12;
    int x = 24;
    mvprintw(y, x-2, "> GAME START");
    mvprintw(y+2, x, "GAME INFO");
    mvprintw(y+4, x, "EXIT");

    refresh();
    while(1){
        int k = keyControl();
        switch(k){
            case UP: {
                if (y > 12){
                    mvprintw(y, x-2, " ");
                    mvprintw(--y, x-2, ">");
                }
                break;
            }
            case DOWN: {
                if (y < 16){
                    mvprintw(y, x-2, " ");
                    mvprintw(++y, x-2, ">");
                }
                break;
            }
            case SUBMIT: {
                return y-12;
            }
        }
        refresh();
    }
}
```

	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16


국민대학교
소프트웨어학부
C++프로그래밍

4.2 Info Draw / 개발: 양지선

- titleDraw 함수에서 Game Info 에 해당하는 값이 return 되었다면 게임 정보를 출력하기 위해 사용되는 함수이다.

```
// 게임 정보 화면
void snake::infoDraw(){
    clear();
    printf("\n\n\n\n\n");
    printf("                < GAME INFORMATION >                \n\n");
    printf(" 1. Use the arrow keys (<,>,v,>) \n");
    printf(" 2. Snake cannot be moved in the opposite direction of the proceeding direction\n");
    printf("    (Failed to move to tail direction of snake)\n");
    printf(" 3. Snake cannot pass its body\n");
    printf(" 4. Snake cannot pass through the wall\n");
    printf(" 5. The movement of the snake in the head direction is shifted by tick\n");
    printf(" 6. Tail increases by 1 when snake acquires a Grow Item\n");
    printf(" 7. If you acquire a Poison Item for a snake, the tail is reduced by 1\n");
    printf(" 8. Snake fails if the body length is less than 3\n\n");
    printf("                Press the <Spacebar> key to continue");
    border('|','|','|','|','|','|','|','|','|','|','|','|','|','|','|','|','|','|','|','|');

    refresh();
    while(1){
        if (keyControl() == SUBMIT){
            clear();
            start();
        }
    }
}
```

	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16


국민대학교
소프트웨어학부
C++프로그래밍

4.3 Stage Select / 개발: 박태범

- titleDraw 함수에서 Game Start 에 해당하는 값이 return 되었다면 원하는 스테이지를 선택할 수 있도록 만들었다.

```
// 스테이지 선택 화면
int snake::stageSelect(){
    int x = 24;
    int y = 6;
    clear();
    printf("\n\n");
    printf("                [ Stage Select ]\n\n");
    mvprintw(y, x-2, "> Stage 1");
    mvprintw(y+2, x, "Stage 2");
    mvprintw(y+4, x, "Stage 3");
    mvprintw(y+6, x, "Stage 4");
    mvprintw(y+8, x, "Back");

    while(1){
        int n = keyControl();
        switch(n){
            case UP: {
                if(y>6){
                    mvprintw(y, x-2, " ");
                    mvprintw(--(y), x-2, ">");
                }
                break;
            }
            case DOWN:{
                if(y<14){
                    mvprintw(y, x-2, " ");
                    mvprintw(++(y), x-2, ">");
                }
                break;
            }
            case SUBMIT:{
                return y-6;
            }
        }
    }
}
```

 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

5. gLoop / 개발: 박태범

- 실제 게임 실행을 위해 실행되는 함수이다. 앞서 선택한 stage 를 인자로 받는다.
- 게임에 필요한 기본적인 초기값을 지정해준다.
- 미리 만들어둔 Map 을 직접 수정하면 재사용이 불가능하기 때문에 tmpMap 으로 복사하여 실행한다.
- 미션을 완료해 게임이 종료되면 다음 스테이지로 넘어갈 수 있게 구현하였다.

```
// gameLoop
void snake::gLoop(int mapCode){
    int x, y;
    nodelay(stdscr, true);
    gItemCnt = 0;
    pItemCnt = 0;
    passGateCnt = 0;
    maxLength = 3;
    currentLength = 3;
    growthItemTime=100;
    poisonItemTime=65;
    gateTime = 120;
    growthItemX = 0;
    growthItemY = 0;
    poisonItemX = 0;
    poisonItemY = 0;
    gateX1 = 0;
    gateY1 = 0;
    gateX2 = 0;
    gateY2 = 0;
    nTail=2;
    _x = 0;
    _y = -1;
    tailX[1] = 0;
    tailY[1] = 0;
    gameOver = false;
    goalGItem = rand()%4+1;
    goalPItem = rand()%4+1;
    goalPassGate = rand()%3+1;
    goalMaxLength = rand()%3+4;

    // mapCode에 맞는 map을 tmpMap에 복사
    if(mapCode == 0){
        memcpy(tmpMap, map1, sizeof(tmpMap));
    }
    else if(mapCode == 1){
        memcpy(tmpMap, map2, sizeof(tmpMap));
    }
    else if(mapCode == 2){
        memcpy(tmpMap, map3, sizeof(tmpMap));
    }
    else if(mapCode == 3){
        memcpy(tmpMap, map4, sizeof(tmpMap));
    }
}
```

```
}
while(!(gameOver || missionCheck())){
    drawMap(&x, &y);
    scoreBoard();
    move(&x, &y);
    growthItem();
    poisonItem();
    usleep(200000);
    openGate();
}

// gameOver 시 실행
if (gameOver){
    nodelay(stdscr, false);
    int menuCode = failDraw();
    if (menuCode == 0){
        snake::gLoop(mapCode);
    }
    else if (menuCode == 2){
        endwin();
    }
}

// mission clear 시 실행
if (missionCheck()){
    nodelay(stdscr, false);
    // 다음 스테이지
    if (mapCode < 3){
        stageClear();
        snake::gLoop(mapCode+1);
    }
    // 클리어
    else{
        victoryDraw();
    }
}
}
```

	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16


5.1 Draw Map / 개발: 양지선

- tmpMap[26][51] 값에 따라 출력을 다르게 하여 숫자로 구성된 맵을 게임 내 사용자가 보는 맵으로 표현하였다. '0'은 공백으로 snake 가 지나다닐 수 있는 길이다. '1'은 Wall, '2'는 Immune Wall 로 각각 '#'과 '*'로 표현하였다. '3'은 snake 의 head 로 'O'로 표현하였다. '4'는 snake 의 body 로 'o' 표현하였다. '5'는 Growth Item, '6'은 Poison Item 으로 각각 'G'와 'P'로 '7'은 Gate 로 '!'를 이용하여 표현하였다. 여기서 각 요소들의 명확한 구분을 위해 Wall 과 Immune Wall 은 흰색, Growth Item 은 파란색, Poison Item 빨간색, Gate 는 노란색으로 표현하였다.

```
// map 그리기
void snake::drawMap(int* x, int* y){
    clear();
    int h, w;

    for(h=0; h<26; h++){
        for(w=0; w<51; w++){
            char tmp = tmpMap[h][w];
            if (tmp == '0'){
                printf(" ");
            }
            else if (tmp == '1'){
                attron(COLOR_PAIR(7));
                printf("#");
                attroff(COLOR_PAIR(7));
            }
            else if (tmp == '2'){
                attron(COLOR_PAIR(7));
                printf("*");
                attroff(COLOR_PAIR(7));
            }
            else if (tmp == '3'){
                *x = h;
                *y = w;
                tailX[0] = h;
                tailY[0] = w;
                if (tailX[1] == 0 && tailY[1] == 0){
                    tailX[1] = h;
                    tailY[1] = w+1;
                    tailX[2] = h;
                    tailY[2] = w+2;
                }
                printf("O");
            }
        }
    }
}
```

```
        else if (tmp == '4'){
            printf("o");
        }
        else if (tmp == '5'){
            attron(COLOR_PAIR(5));
            printf("G");
            attroff(COLOR_PAIR(5));
        }
        else if (tmp == '6'){
            attron(COLOR_PAIR(4));
            printf("P");
            attroff(COLOR_PAIR(4));
        }
        else if (tmp == '7'){
            attron(COLOR_PAIR(2));
            printf("!");
            attroff(COLOR_PAIR(2));
        }
    }
    printf("\n");
}
```

 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

5.2 ScoreBoard / 개발: 양지선


- ScoreBoard 에는 현재 플레이어의 기록, MissionBoard 에는 스테이지의 목표를 표시하도록 하였고 미션을 성공하면 해당 미션 옆에 V 표시가 나타나도록 구현하였다.

```
// 점수판
void snake::scoreBoard(){

    mvprintw(0, 60, "*****");
    mvprintw(1, 60, "*-----SCORE BOARD-----*");
    mvprintw(2, 60, "** Score : %d          *", snake::getScore());
    mvprintw(3, 60, "** B : %d / %d          *", currentLength, maxLength);
    mvprintw(4, 60, "** + : %d              *", gItemCnt);
    mvprintw(5, 60, "** - : %d              *", pItemCnt);
    mvprintw(6, 60, "** G : %d              *", passGateCnt);
    mvprintw(7, 60, "*****");

    mvprintw(10, 60, "*****");
    mvprintw(11, 60, "*-----MISSION BOARD-----*");
    mvprintw(12, 60, "** B : %d              *", goalMaxLength);
    mvprintw(13, 60, "** + : %d              *", goalGItem);
    mvprintw(14, 60, "** - : %d              *", goalPItem);
    mvprintw(15, 60, "** G : %d              *", goalPassGate);
    mvprintw(16, 60, "*****");

    if (maxLength >= goalMaxLength){
        mvprintw(12, 75, "V");
    }
    if (gItemCnt >= goalGItem){
        mvprintw(13, 75, "V");
    }
    if (pItemCnt >= goalPItem){
        mvprintw(14, 75, "V");
    }
    if (passGateCnt >= goalPassGate){
        mvprintw(15, 75, "V");
    }
    refresh();
}
```


 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

5.3 Growth Item / 개발: 박태범

- 일정 시간이 지나면 tmpMap 의 값이 '0'인 곳에 Growth Item('5')이 생성되도록 하였다.

```
// growthItem 생성
void snake::growthItem(){

    int prevGrowthItemX = growthItemX;
    int prevGrowthItemY = growthItemY;

    if (growthItemTime >= 100){

        growthItemTime = 0;
        // item 제거
        if (tmpMap[prevGrowthItemX][prevGrowthItemY] == '3' || tmpMap[prevGrowthItemX][prevGrowthItemY] == '4'){
        }
        else if (!(prevGrowthItemX==0 && prevGrowthItemY==0)){
            tmpMap[prevGrowthItemX][prevGrowthItemY] = '0';
        }
        while (1){
            growthItemX = rand()%21+2;
            growthItemY = rand()%46+2;
            if (tmpMap[growthItemX][growthItemY] == '0'){
                break;
            }
        }
        tmpMap[growthItemX][growthItemY] = '5';
    }
    else{
        growthItemTime++;
    }
}
```

5.4 Poison Item / 개발: 박태범

- 위와 마찬가지로 일정 시간이 지나면 tmpMap 의 값이 '0'인 곳에 Poison Item('6')이 생성되도록 하였다.

```
// poisonItem 생성
void snake::poisonItem(){

    int prevPoisonItemX = poisonItemX;
    int prevPoisonItemY = poisonItemY;

    if (poisonItemTime >= 80){

        poisonItemTime = 0;
        //item 제거
        if (tmpMap[prevPoisonItemX][prevPoisonItemY] == '3' || tmpMap[prevPoisonItemX][prevPoisonItemY] == '4'){
        }
        else if (!(prevPoisonItemX==0 && prevPoisonItemY==0)){
            tmpMap[prevPoisonItemX][prevPoisonItemY] = '0';
        }
        while (1){
            poisonItemX = rand()%21+2;
            poisonItemY = rand()%46+2;
            if (tmpMap[poisonItemX][poisonItemY] == '0'){
                break;
            }
        }
        tmpMap[poisonItemX][poisonItemY] = '6';
    }
    else{
        poisonItemTime++;
    }
}
```


	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

국민대학교
소프트웨어학부
C++프로그래밍

5.5 Gate / 개발: 박태범

- Gate 는 일정 시간이 지나면 tmpMap 의 값이 '1'인 곳에 2 개의 Gate('7')가 생성되도록 하였다.

```
// 게이트 생성
void snake::openGate(){
    int prevGateX1 = gateX1;
    int prevGateY1 = gateY1;
    int prevGateX2 = gateX2;
    int prevGateY2 = gateY2;
    if (gateTime >= 120){
        if (!(prevGateX1==0 && prevGateY1==0)){
            tmpMap[prevGateX1][prevGateY1] = '1';
        }
        if (!(prevGateX2==0 && prevGateY2==0)){
            tmpMap[prevGateX2][prevGateY2] = '1';
        }
        while (1){
            gateX1 = rand()%25;
            gateY1 = rand()%50;
            if (tmpMap[gateX1][gateY1] == '1'){
                break;
            }
        }
        while (1){
            gateX2 = rand()%25;
            gateY2 = rand()%50;
            if (tmpMap[gateX2][gateY2] == '1' && (gateX2 != gateX1 || gateY2 != gateY1)){
                break;
            }
        }
        tmpMap[gateX2][gateY2] = '7';
        tmpMap[gateX1][gateY1] = '7';
        gateTime = 0;
    }
    else{
        gateTime++;
    }
}
```

 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

5.6 Move / 개발: 박태범

- Snake의 이동을 담당하는 부분이다. 우선 한 틱 이동을 하게되면 기존 Snake의 가장 끝부분은 0으로 바꿔주어야 한다. 그 후 Snake의 Head를 제외한 tailX[i], tailY[i] 좌표를 tailX[i-1], tailY[i-1]의 좌표로 변경하여 Tail을 한 칸씩 이동시키고 마지막으로 Head의 값을 사용자의 입력에 맞게 변경된 위치에 놓아주는 방식으로 구현하였다.

```
void snake::move(int* x, int* y){
    // snake body 이동
    tmpMap[tailX[nTail]][tailY[nTail]] = '0';
    for (int i=nTail; i>0; i--){
        tailX[i] = tailX[i-1];
        tailY[i] = tailY[i-1];
        tmpMap[tailX[i]][tailY[i]] = '4';
    }


    int tmp = getch();

    switch(tmp){
        case KEY_UP:
            _x = -1;
            _y = 0;
            break;
        case KEY_DOWN:
            _x = 1;
            _y = 0;
            break;
        case KEY_RIGHT:
            _x = 0;
            _y = 1;
            break;
        case KEY_LEFT:
            _x = 0;
            _y = -1;
            break;
    }

    *x += _x;
    *y += _y;

    char mapObject = tmpMap[*x][*y];
```

```
// head 이동
tmpMap[*x][*y] = '3';
```

	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

5.6.1 Gate 통과 / 개발: 박태범

- Gate 통과시에도 Head 의 좌표와 이동방향만 설정해주면 위의 코드로 이동이 가능하다. Gate 를 빠져나오는 조건에 맞게 방향과 좌표를 지정해주었다.

```

// gate1 in -> gate2 out
if (*x == gateX1 && *y == gateY1){
    passGateCnt++;
    if ((120-gateTime) < nTail){
        gateTime = 119 -nTail;
    };
    if (tmpMap[gateX2+_x][gateY2+_y] == '0'){
        *x = gateX2+_x;
        *y = gateY2+_y;
    }
    else if (tmpMap[gateX2][gateY2-1] == '0'){
        _x = 0;
        _y = -1;
        *x = gateX2;
        *y = gateY2-1;
    }
    else if (tmpMap[gateX2-1][gateY2] == '0'){
        _x = -1;
        _y = 0;
        *x = gateX2-1;
        *y = gateY2;
    }
    else if (tmpMap[gateX2][gateY2+1] == '0'){
        _x = 0;
        _y = 1;
        *x = gateX2;
        *y = gateY2+1;
    }
    else if (tmpMap[gateX2+1][gateY2] == '0'){
        _x = +1;
        _y = 0;
        *x = gateX2+1;
        *y = gateY2;
    }
}

// gate2 in -> gate1 out
if (*x == gateX2 && *y == gateY2){
    passGateCnt++;
    if ((120-gateTime) < nTail){
        gateTime = 119 -nTail;
    };
    if (tmpMap[gateX1+_x][gateY1+_y] == '0'){
        *x = gateX1+_x;
        *y = gateY1+_y;
    }
    else if (tmpMap[gateX1][gateY1-1] == '0'){
        _x = 0;
        _y = -1;
        *x = gateX1;
        *y = gateY1-1;
    }
    else if (tmpMap[gateX1-1][gateY1] == '0'){
        _x = -1;
        _y = 0;
        *x = gateX1-1;
        *y = gateY1;
    }
    else if (tmpMap[gateX1][gateY1+1] == '0'){
        _x = 0;
        _y = 1;
        *x = gateX1;
        *y = gateY1+1;
    }
    else if (tmpMap[gateX1+1][gateY1] == '0'){
        _x = 1;
        _y = 0;
        *x = gateX1+1;
        *y = gateY1;
    }
}

```

	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

5.6.2 Get Growth Item / 개발: 박태범

- Growth Item 을 획득하면 총 길이를 1 증가시키고 tailX, tailY 에 좌표를 추가시킨다.

```
// get growthItem
if (mapObject == '5'){
    gItemCnt++;
    growthItemTime = 100;
    nTail++;
    if (nTail+1 > maxLength){
        maxLength = nTail+1;
    }
    currentLength = nTail+1;
    tailX[nTail] = *x-_x;
    tailY[nTail] = *y-_y;
    tmpMap[tailX[nTail]][tailY[nTail]] = '4';
}
```

5.6.3 Get Poison Item / 개발: 박태범

- Poison Item 을 획득하면 총 길이를 1 감소시킨다.

```
// get poisonItem
if (mapObject == '6'){
    pItemCnt++;
    tmpMap[tailX[nTail]][tailY[nTail]] = '0';
    poisonItemTime = 80;
    nTail--;
    currentLength--;
    if (nTail <=1){
        gameOver = true;
    }
}
```

	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

국민대학교
소프트웨어학부
C++프로그래밍


5.7 Fail Draw / 개발: 박태범

- Snake 의 Head 가 Wall('1')이나 Tail('4')에 닿으면 게임이 종료되는데 Gameover 값이 True 일 때 출력되는 화면이다.
- 좌표에 따라 재시작, 종료를 선택할 수 있게 구현하였다.

```
// 실패 화면
int snake::failDraw(){
    clear();

    attron(COLOR_PAIR(4));
    printf("\n\n\n\n\n\n\n");
    printf("                                GAME OVER                                \n");
    attroff(COLOR_PAIR(4));
    printf("                                ( You can play the game again by pressing Restart )");
    int y = 12;
    int x = 27;
    mvprintw(y, x-2, "> RESTART");
    mvprintw(y+2, x, "EXIT");

    refresh();
    while(1){
        int k = keyControl();
        switch(k){
            case UP: {
                if (y > 12){
                    mvprintw(y, x-2, " ");
                    mvprintw(--(--y), x-2, ">");
                }
                break;
            }
            case DOWN: {
                if (y < 14){
                    mvprintw(y, x-2, " ");
                    mvprintw(++(++y), x-2, ">");
                }
                break;
            }
            case SUBMIT:{
                return y-12;
            }
        }
        refresh();
    }
}
```

 <div> 국민대학교 소프트웨어부 C++프로그래밍 </div>	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

5.8 Stage Clear / 개발: 양지선

- 게임이 종료되면 Mission Check 함수에서 미션이 완료되었는지 확인하고 미션을 성공했으면 출력되는 화면이다.

```
bool snake::missionCheck(){
    if (gItemCnt >= goalGItem && pItemCnt >= goalPItem && passGateCnt >= goalPassGate && maxLength >= goalMaxLength){
        return true;
    }
    return false;
}
```

```
// 스테이지 클리어
void snake::stageClear(){
    clear();
    printf("\n\n\n\n\n\n\n");


    attron(COLOR_PAIR(3));
    printf("                                Congratulation!\n\n");
    attroff(COLOR_PAIR(3));
    printf("                                YOU CLEAR THIS STAGES\n\n");
    printf("                                You Got %d Score \n\n\n\n", getScore());
    printf("                                Press Any Key to Continue");
    refresh();
    getch();
}
```

5.9 Victory / 개발: 양지선

- 마지막 스테이지를 클리어했을 때 출력되는 화면이다.

```
// 클리어 화면
void snake::victoryDraw(){
    clear();
    printf("\n\n\n\n\n\n\n");

    attron(COLOR_PAIR(3));
    printf("                                Congratulation!\n\n");
    attroff(COLOR_PAIR(3));
    printf("                                CLEAR ALL THE STAGES\n\n\n\n\n\n");
    printf("                                Press the <Spacebar> key to Continue");
    refresh();
    while(1){
        if (keyControl() == SUBMIT){
            clear();
            start();
        }
    }
}
```

 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

2.2.3 활용/개발된 기술

작성요령 (10 점)

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.

1. <ncurses.h>: GUI 기반으로 게임을 만들기 위해 사용하였다.
2. <iostream>: 기본적인 입,출력을 위해 사용하였다.
3. <string.h>: String 형식의 문자를 입,출력하기 위해 사용하였다.
4. <unistd.h>: Snake 의 속도를 조절하기 위해 사용하였다.

2.2.4 현실적 제한 요소 및 그 해결 방안

작성요령 (5 점)

제안된 프로젝트의 단계 별 수행에 있어, 제한 요소를 찾아 작성한다. 해당 제한 요소를 해결하기 위해서 어떤 방법으로 해결하였는지 작성한다.


Wall, Gate, Item, Snake를 출력할 때 구분이 용이하게 특수문자로 표시하고 싶었으나 너무 많은 부분을 바꿔야했기 때문에 각각 요소에 색을 지정하여 구분이 가능하게 만들었다.

2.2.5 결과물 목록

작성요령 (5 점)

결과물 목록을 작성한다. 목록은 제출하는 파일과 각 파일의 역할을 간략히 설명한다.

파일명	역할
Main.cpp	Snake Game 을 실행하는 파일
Snake.h	Snake.cpp 파일의 헤더파일
Snake.cpp	Snake game 의 핵심 로직이 구현된 파일
Makefile	컴파일을 하기 위한 Makefile

 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

3 자기평가

작성요령 (5 점)

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.

[양지선]

게임 내 화면 디자인을 맡았다.

프로젝트를 수행하면서 제일 어려웠던 점은 클래스나 상속과 같은 수업 진도가 나가고 있는 부분을 바로 응용해 코드를 짜는 부분이 어려웠던 것 같다. 하지만 프로젝트를 진행하면서 수업 시간에 이해가 안가던 부분에 대한 이해를 높일 수 있어 좋은 시간었던 것 같다. 또한 같은 팀원의 코드를 보고 배울 수 있던 점도 많았다. 차후 이 프로젝트 운영에 대해 바라는 점은 처음 snake game 을 구현해서 제출하러했을 때 막막했던 느낌이 강했기에 본격적인 게임 구현 전에 팀원과 같이 목표나 계획 설정에 대한 예비보고서를 작성하는 시간이 있으면 좋을 것 같다.


[박태범]

본 프로젝트에서 전반적인 게임의 기능구현을 담당하였다.

이러한 프로젝트를 조금 더 경험해본 입장으로 설계에 더 신경을 써야했는데 클래스에 대한 숙련도가 부족해 결과적으로 나만 알아볼 수 있는 코드가 된 것 같아 아쉬움이 많이 남았다. 또한 기능을 snake.cpp 파일 하나에서 구현하다보니 github 도 잘 활용하지 못한 것 같다. 그래도 프로젝트에서 요구하는 기능들은 모두 구현할 수 있어 뿌듯함을 느꼈다.

4 참고 문헌

번호	종류	제목	출처	발행년도	저자	기타
1	웹페이지	C++ Reference	https://cplusplus.com/reference/			

 국민대학교 소프트웨어학부 C++ 프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

5 부록

작성요령 (15 점)

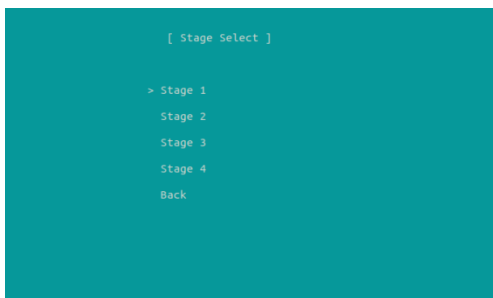
프로젝트의 결과물을 사용하기 위한 방법에 대해서 작성하세요.

5.1 사용자 매뉴얼

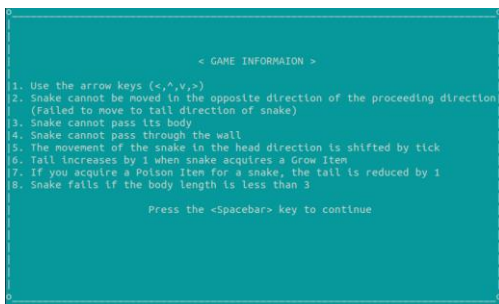
1) 시작화면에서 방향키(↑,↓)와 space bar 를 이용하여 'GAME START', 'GAME INFO', 'EXIT' 중 선택하여 이동할 수 있다.



– GAME START : stage 를 선택할 수 있는 화면으로 넘어간다.



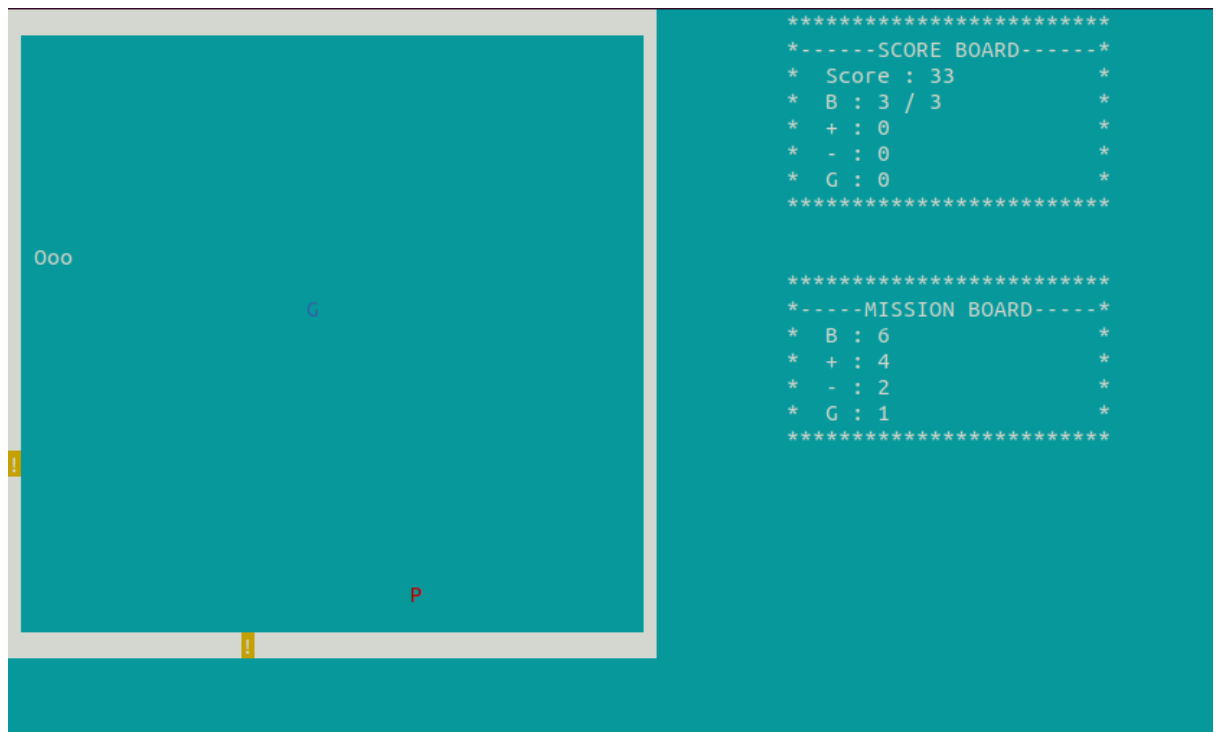
– GAME INFO: 게임 조작법과 룰이 적혀있는 화면으로 넘어간다.



– EXIT: 게임을 종료한다.

 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

- 2) GAME START 를 클릭한 후 Stage 를 선택하면 게임이 시작된다.
- 3) 왼쪽 창은 게임 진행 창이며 오른쪽 상단 'SCORE BOARD'는 현재 Score 를 보여주며 오른쪽 하단 'MISSION BOARD'는 stage clear 를 위한 미션을 보여준다.
 - Score : (최고 길이*10 + 아이템 획득 횟수*5 + 게이트 통과 횟수*20 + 현재 길이) 로 구성된 통합 점수이다.
 - B : 해당 스테이지에서의 snake 의 최고 길이와 현재 길이를 나타낸다.
 - + : 'G' 아이템을 획득한 횟수이다.
 - - : 'P' 아이템을 획득한 횟수이다.
 - G : Gate 를 통과한 횟수이다.
- 4) snake 는 방향키(←,↑,↓,→)를 이용하여 조작할 수 있다.
- 5) 'G' 아이템을 획득하면 snake 의 길이가 1 늘어난다. 반대로 'P' 아이템을 획득하면 snake 의 길이가 1 작아진다.
- 6) 테두리 흰색 벽에 부딪히거나 snake head 의 반대방향으로 조작하면 게임은 종료된다. 또한 snake 의 길이가 3 보다 작아져도 종료된다.
- 7) 벽 중 노란색 벽은 Gate 로 해당 Gate 로 진입하면 또 다른 Gate 로 진출할 수 있다.
- 8) 모든 mission 을 달성하면 stage clear!



 국민대학교 소프트웨어학부 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	팀C	
	Confidential Restricted	Version 1.2	2022-JUN-16

5.2 설치 방법

- 1) snake.zip 파일을 다운로드 후 압축을 해제한다.
- 2) 터미널에 **make** 를 입력하여 컴파일한다.
- 3) ./snake 를 입력하여 컴파일된 snake 를 실행하면 SnakeGame 의 첫 화면이 등장한다.