




국민대학교  
소프트웨어융합대학  
소프트웨어학부

# C++프로그래밍 프로젝트

프로젝트 명	Snake game
팀 명	HelloSnake
문서 제목	결과보고서

Version	2.0
Date	2021-Jun-14

팀원	김규리(20203031)
	민경서(20203060)

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>C++프로젝트</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Snake game	
	<b>팀 명</b>	<b>HelloSnake</b>	
	Confidential Restricted	Version 2.0	2020-JUN-14


#### 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 "HelloSnake"의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 "HelloSnake"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

### 문서 정보 / 수정 내역

<b>Filename</b>	Snake(최종).doc
<b>원안작성자</b>	김규리, 민경서
<b>수정작성자</b>	김규리, 민경서

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2020-06-13	민경서	1.0	최초 작성	전 내용
2020-06-13	김규리	2.0	내용 수정	전 내용 수정

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>C++프로젝트</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Snake game	
	<b>팀 명</b>	<b>HelloSnake</b>	
	Confidential Restricted	Version 2.0	2020-JUN-14

## 목 차

- 1 개요 4
- 2 개발 내용 및 결과물 5
  - 2.1 목표 5
  - 2.2 개발 내용 및 결과물 7
    - 2.2.1 개발 내용 7
    - 2.2.2 시스템 구조 및 설계도 8
    - 2.2.3 활용/개발된 기술 16
    - 2.2.4 현실적 제한 요소 및 그 해결 방안 17
    - 2.2.5 결과물 목록 18
- 3 자기평가 19
- 4 참고 문헌 20
- 5 부록 20
  - 5.1 사용자 매뉴얼 20
  - 5.2 설치 방법 21

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>C++ 프로젝트</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Snake game	
	<b>팀 명</b>	HelloSnake	
	Confidential Restricted	Version 2.0	2020-JUN-14

## 1 개요

### 평가기준 (10 점)


프로젝트를 완성하기 위해 사용한 개발 방법을 기술하세요.

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

C++ 프로그래밍 기말 프로젝트 과제는 ncurses 라이브러리를 사용하여 Snake Game 를 구현하는 것이다. ncurses 라이브러리를 획득하기 위해 Ubuntu 환경에서 “sudo apt-get install libncurses-dev libncursesw5-dev” 명령어를 실행시켜 ncurses 라이브러리를 획득한다.

Snakegame 을 완성하기 위해 처음에는 만나서 계획을 짰고 그 이후에는 zoom 과 구글문서, 카카오톡을 통해 진행하였다. 게임을 제작하는 데에 오랜 시간이 걸릴 것 같아 저희는 매주 3-4 번의 회의와 zoom 을 이용한 실시간 코딩을 통해 게임을 완성하려고 노력했다. 회의를 통해서 어떤 방식을 통해 게임을 개발할 것이고, 어느 부분을 누가 짤 것인지, 어떤 알고리즘을 사용할지에 대한 계획을 세웠다. Zoom 으로 실시간 코딩을 하면서 코드의 문제를 함께 고쳐 나가고 코드를 구글 문서를 통해 공유했다. 각 단계가 구현될 때마다 github 에 push 해 코드를 잃어버리지 않고 최종 코드를 완성할 수 있게 되었다.

개발을 진행하면서 저희는 Vector, ncurses, ctime 라이브러리를 사용했고 검색을 통해서 설치하고 사용했다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>C++프로젝트</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Snake game	
	<b>팀 명</b>	HelloSnake	
	Confidential Restricted	Version 2.0	2020-JUN-14

## 2 개발 내용 및 결과물

### 2.1 목표

#### 작성요령 (10 점)

프로젝트의 목표를 기술하세요. 각 단계별 목표를 구체적으로 쓰세요.

적용단계	내용	적용 여부
1 단계	Map 의 구현	적용
2 단계	Snake 표현 및 조작	적용
3 단계	Item 요소의 구현	적용
4 단계	Gate 요소의 구현	적용
5 단계	점수 요소의 구현	적용

#### 1 단계 : Map 의 구현

- User의 터미널에 2차원 배열로 된 Snake Map을 표시한다.
- ncurses라이브러리를 사용해 Wall, Immune Wall을 잘 구분해서 map을 구현한다.
- 게임의 진행 상황인 Mission, score board를 구현한다.

#### 2 단계 : Snake 표현 및 조작

- Map 위에 Snake를 표시하고 User가 방향키로 움직일 수 있게 한다.
- Snake의 머리가 벽에 닿으면 게임이 종료되도록 하고, 게임 보드를 벗어나면 종료되도록 한다.
- Snake는 일정 시간(틱)에 의해 이동한다.


#### 3 단계 : growth, poison item 의 출현

- Map위에 임의의 위치에 growth item, poison item 이 랜덤하게 출현되도록 하였다.
- 5초간 item을 획득하지 못할 시에 item이 재생성되는 기능을 추가했다.
- Growth item : 몸의 길이 1증가
- Poison item : 몸의 꼬리 길이 1감소
- Item 을 획득하면 위치가 다시 랜덤으로 바뀐다.

#### 4 단계 : Gate 요소의 구현

- Wall 임의의 위치에 한 쌍의 gate가 출현하도록 한다.
- Gate는 올바른 방향으로 통과하고 재생성되는 과정을 구현하도록 한다.

#### 5 단계 : 점수 요소의 구현

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>C++프로젝트</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Snake game	
	<b>팀 명</b>	<b>HelloSnake</b>	
	Confidential Restricted	Version 2.0	2020-JUN-14

- 게임이 끝나면 mission board 위에 snake 의 현재 길이와 뱀의 최대 길이, 획득한 growth, poison item, gate의 사용 횟수를 표시한다.
- Mission을 통과할 때마다 다음 단계의 Map으로 바뀌는 기능을 추가하도록 한다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>C++프로젝트</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Snake game	
	<b>팀 명</b>	<b>HelloSnake</b>	
	Confidential Restricted	Version 2.0	2020-JUN-14

## 2.2 개발 내용 및 결과물

### 2.2.1 개발 내용

#### 작성요령 (10 점)

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

저희는 약 3 주차에 걸쳐 프로젝트를 계획하고 실행했습니다.

매주 해야 할 목표를 정하고 코드를 짜고 완성하였습니다.

1 주차 : Map 구현 및 윈도우 나누기, Snake 표시 및 움직임 구현

Ncurses 라이브러리를 사용하여 터미널에 표시될 게임윈도우, 점수윈도우, 미션윈도우로 분할한다.  
map에서 wall은 게임이 시작하자마자 형성되고, 벽의 각 꼭짓점은 immune wall의 상태로 변경한다.

2 주차 : growth, poison item 출현, gate 출현

Map 위에 growth, poison item 을 출력하고 한 쌍의 gate 를 출력한다. Gate 는 각 스테이지마다 한 쌍 씩 생기고, map 의 wall1 중에 임의의 두 곳을 랜덤으로 선택하여 gate 의 위치를 선정한다.

3 주차 : score, mission 표시하기와 게임 종료 구현


각 스테이지에 맞는 mission 을 정하고 미션을 클리어 할 때마다 단계가 올라가고 맵이 바뀌며, 조건을 완료하면 게임 오버가 된다.

Game Over :

1. 뱀의 몸 길이 3 미만
2. 뱀 자신의 몸에 닿을 때
3. 벽에 부딪힐 때
4. poison item 을 특정 개수 이상으로 먹었을 때

Game Clear : Mission clear

+ : 여러 오류를 수정하고 최종 완성

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>C++ 프로젝트</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Snake game	
	<b>팀 명</b>	<b>HelloSnake</b>	
	Confidential Restricted	Version 2.0	2020-JUN-14

## 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.

Step 1. Map 의 구현

-map.h

단계별로 달라질 map 들을 넣어놨습니다.

-snake.cpp

Map 을 표기하기 위해 먼저 윈도우 공간을 구성했습니다. Ncurses 를 사용하여, game, score, mission 으로, 세 가지의 윈도우 구성하였습니다. game 에서 게임이 이루어지고, score 에서는 점수가 보여집니다. mission 에서는 미션 목록이 보여지는 공간입니다.

```
WINDOW *game = newwin(y+2, x-38, 0, 0); //height, width, startY, startX
```

```
wbkgd(game, COLOR_PAIR(1));
```

```
wattron(game, COLOR_PAIR(1));
```

```
wborder(game, '|','|','-','-','+','+','+','+');
```

```
nodelay(game, TRUE);
```

```
keypad(game, TRUE);
```

또, 차후의 snake 의 움직임과 지금의 벽 위치를 vector 좌표로 지정할 수 있는 Vector 클래스를 생성했습니다.

```
class Vector
```

```
{
```

```
public:
```

```
Vector() : x(0), y(0){
```

```
Vector(int x, int y) {setX(x); setY(y);}
```

```
private:
```

```
int x,y;
```

```
}
```



 <b>국민대학교</b> <b>소프트웨어학부</b> <b>C++프로젝트</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Snake game	
	<b>팀 명</b>	<b>HelloSnake</b>	
	Confidential Restricted	Version 2.0	2020-JUN-14

Snake 클래스 속 init 함수로 map.h 에 있는 텍스트 형태의 map 모양을 가지고 온 뒤에 check\_wall 라는 함수를 이용해 벽 위치를 벡터배열로 생성합니다. 변수 x, y 를 통해 좌표값을 설정할 수 있습니다.

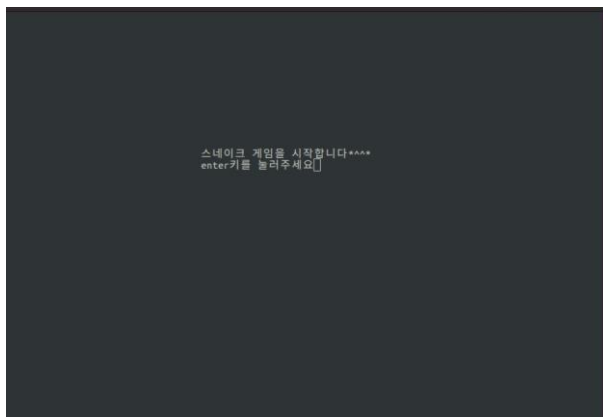
```
void check_wall(int map[30][60]) {
    for(int i=0; i<30; i++){
        for(int j=0; j<60; j++){
            if(map[i][j]==1){
                wall1[countW++] = Vector(j,i);
            }
        }
    }
}
```

```
setlocale(LC_ALL, "ko_KR.utf8");
```

```
setlocale(LC_CTYPE, "ko_KR.utf8"); //한글 출력 설정
```


시작 화면에 한글을 표시하기 위하여, - 설정을 한글 출력으로 해주고 -lncursesw 로 컴파일합니다.

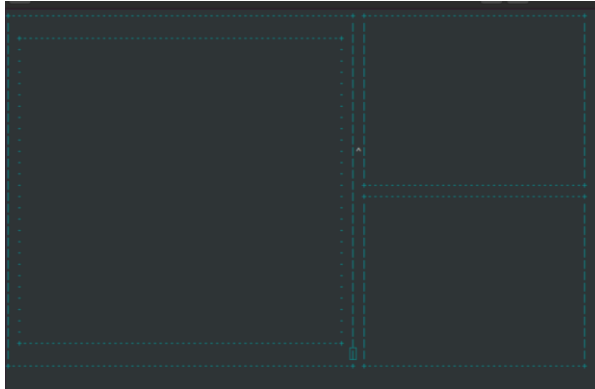
mwprintw 함수를 사용하여 간단한 내용을 출력합니다.



create 라는 함수를 통해 game window 위에 map 을 만들었습니다.

get\_table 은 cstring 의 memset 함수를 통해 원하는 크기만큼 세팅합니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>C++ 프로젝트</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Snake game	
	<b>팀 명</b>	<b>HelloSnake</b>	
	Confidential Restricted	Version 2.0	2020-JUN-14



-> 1 단계 Map 모양

## Step 2. Snake 표현 및 조작

snake 클래스 속 `get_table` 함수에 map 을 구현하면서 머리와 몸 좌표를 나준 뒤에, `create` 함수로 map 구현할 때 같이 원하는 모양으로 출력해준다. `getBodySize` 함수를 통해 snake 의 몸 길이를 받습니다.

-snake.cpp: class Snake : `get_table` 함수

```
table[body[0].getY()*width+body[0].getX()] = 'h';

for(unsigned int i=1; i<body.size(); ++i)

    table[body[i].getY()*width+body[i].getX()] = 'b';
```

-snake.cpp: `create` 함수

```
case 'h':

    mvwprintw(game, 1+y,1+x, "O");

    break;


case 'b':

    mvwprintw(game, 1+y,1+x, "o");

    break;
```

움직임은 snake 클래스의 `make_move()`를 이용하여 bool 값인 `exit` 가 true 이지 않는 이상 계속 진행한다. 그리고 snake 의 머리와 몸 부분이 이동하게 됩니다.

그리고 `set_direction` 함수를 통해 이동방향을 설정해줍니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>C++프로젝트</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Snake game	
	<b>팀 명</b>	<b>HelloSnake</b>	
	Confidential Restricted	Version 2.0	2020-JUN-14

```
void set_direction(int d){
    switch(d){
        case 0:direction = Vector(0, -1); break;
        case 1:direction = Vector(1, 0); break;
        case 2:direction = Vector(0, 1); break;
        case 3:direction = Vector(-1, 0); break;
    }
}
```

main 부분은 루프를 만들어 입력 받고 방향을 설정해준 뒤 움직이고 game Window 에 띄우는 식으로 만들었습니다.

```
While(true){
    input = wgetch(game);
    proccesInput(game, snake, input);
    if(snake.get_exit())break;
    snake.body_move();
    snake.make_move();
    refresh();
    wrefresh(game);
}
```

STEP3 growth/poison item 구현

<growth, poison item 놓기>



```
void growth_item(int map[30][60]){ // growth 놓기
    startAT = time(NULL);

    int x = rand()%width;
    int y = rand()%height;
    growth = Vector(x,y);
    for(unsigned int i=0;i<body.size();++i)
        if(growth==body[i])growth_item(map);
        if(map[y][x]!=0) growth_item(map);
}

void poison_item(int map[30][60]){ // poison 위치 설정
    startPT = time(NULL);

    int x,y;
    x = rand()%width;
    y = rand()%height;
    poison = Vector(x,y);
    for(unsigned int i=0;i<body.size();++i)
        if(poison==body[i])poison_item(map);

        if(map[y][x]!=0) poison_item(map);
}
```

Growth\_item, Poison\_item : 랜덤으로 좌표를 정해주고 그 좌표에 아이템을 생성하는 함수이다. startTime 은 5 초동안 item 을 획득하지 못하면 item 이 재생성 되도록 하였다. 현재 시간을 갱신해줄 때는 ctime 라이브러리의 time(NULL) 을 이용하였다.

<item 획득 여부 확인>



```
void checkForGrowth(){

    nowAT = time(NULL);

    if(nowAT - startAT > itemT){
        growth_item(map);
    }
    if(body[0]==growth)
    {
        count_growth++; // 수정
        points++;
        if(points>best)best = points;
        growth_item(map);
        increaseBodySize(body.size());
    }
}

void checkForPoison(){ // 길이 줄이고 포인트 감소


    nowPT = time(NULL);

    if(nowPT - startPT > itemT){
        poison_item(map);
    }

    if(body[0]==poison)
    { count_poison++; // 수정
      points--;
      poison_item(map);
      reduceBodySize(body.size());
    }
}
```

checkForGrowth, checkForPoison : snakegame.cpp 파일의 main 문 속 while 문에서 snake 가 item 을 획득했는지 좌표값을 통해 검사하는 함수이다. 계속 nowTime 을 현재 시간으로 갱신해주고, 5 초보다 커질 시엔 item 을 재생성 해주 기능을 구현했다.

<snake size 정하기>

 <div> 국민대학교  소프트웨어학부  C++프로젝트 </div>	결과보고서		
	프로젝트 명	Snake game	
	팀 명	HelloSnake	
	Confidential Restricted	Version 2.0	2020-JUN-14

```

void increaseBodySize(uint i){
    if(i >= body.size())
        body.push_back(body.back());
    else
        body[i] = body[i - 1];
}

void reduceBodySize(uint i){
    if(i >= body.size())
        body.pop_back();
    else
        body[i] = body[i - 1];
}

```

increaseBodySize, reduceBodySize : Item 을 획득할 때, snake size 를 효율적으로 조절하기 위해 구현한 함수이다.

#### STEP4 Gate 요소의 구현

<벽 위치 벡터배열로 생성>


```

void check_wall(int map[30][60]){ // 벽 위치 벡터배열로 생성
    num_1 = 0;
    for(int i=0; i<30; i++){
        for(int j=0; j<60; j++){
            if(map[i][j]==1){
                wall1[num_1++] = Vector(j,i);
            }
        }
    }
}

```

Check\_wall : 벽이 생성되는 위치를 vector 배열로 생성해주는 함수이다. For 문을 통해 map 을 검사해주며 값이 1 인 좌표를 vector 배열에 저장해주는 방식이다.

<gate 생성>

 <div> 국민대학교  소프트웨어학부  C++ 프로젝트 </div>	결과보고서		
	프로젝트 명	Snake game	
	팀 명	HelloSnake	
	Confidential Restricted	Version 2.0	2020-JUN-14

```
void get_gate(Vector wall[]){ //gate 위치 정해주기
    int A = rand()%num_1;
    int B = rand()%num_1;
    if(B == A) get_gate(wall1);
    gate[0] = wall1[A];
    gate[1] = wall1[B];
}
```

get\_gate : 랜덤 난수 생성 기능을 통해 gate 의 양쪽 입구 좌표를 선택하였고, check\_wall 함수를 통해 Vector 배열에 저장해 놓은 값들만 난수를 생성하게 하여 벽에만 gate 가 생기도록 하였다.

<gate 이동함수>

```
void check_gate(){ // 몸이 다 통과할때까지 머리 방향 설정
    if(body[0] == gate[0]){
        body[0] = gate[1];
        count_gate++;
        set_direction(gate_direction(gate[1]));
    }

    else if(body[0] == gate[1]){
        count_gate++;
        body[0] = gate[0];
        set_direction(gate_direction(gate[0]));
    }
}

void change_gate(){ // 다 통과해야 게이트 바뀌도록
    Vector A = direction;
    for(int i=0; i<body.size(); i++){
        A = A + direction;
    }

    if(body[0] == gate[0]+A){
        get_gate(wall1);
    }

    else if(body[0] == gate[1]+A){
        get_gate(wall1);
    }
}
```

Check\_gate : snake 의 몸이 다 통과할 때까지의 몸의 방향을 설정, 한 쪽 gate 를 통과하면 다른 쪽의 gate 로 나올 수 있도록 하는 함수이다.

Change\_gate : snake 의 몸이 gate 를 다 통과해야 새로운 gate 가 다른 곳에 생성되도록 하는 함수이다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>C++ 프로젝트</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Snake game	
	<b>팀 명</b>	<b>HelloSnake</b>	
	Confidential Restricted	Version 2.0	2020-JUN-14

<growth, poison, gate 모든 정보를 전체 배열을 사용하기 편리한 형태로 저장하는 함수>

```
char* get_table(int map[30][60])
{
    std::memset(table, ' ', height*width);

    table[body[0].getY()*width+body[0].getX()] = 'h';
    for(unsigned int i=1; i<body.size(); ++i)
        table[body[i].getY()*width+body[i].getX()] = 'b';

    table[growth.getY()*width+growth.getX()] = 'a';
    table[poison.getY()*width+poison.getX()] = 'x'; //poison

    for(unsigned int i=0; i<30; i++)
    {
        for(int j=0; j<60; j++){
            if(map[i][j]==1) table[i*width+j] = '1' ;
            else if(map[i][j]==2) table[i*width+j] = '2' ;
            else if (map[i][j]==3) table[i*width+j] = '3';
        }
    }
    table[gate[0].getY()*width+ gate[0].getX()] = 'G';
    table[gate[1].getY()*width+ gate[1].getX()] = 'G';

    return table;
}
```

### 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.

window 를 띄우거나, 시각적인 요소들을 사용하기 위해서 Ncurses 를 이용하였습니다.

memset 함수를 이용해 원하는 크기만큼 세팅하기 위해 cstring 을 사용하였습니다.

usleep 함수로 실행 시간을 멈춰놓기 위해 ctime 을 사용하였습니다.

Vector 클래스로 x, y 좌표값을 지정해서 사용하기 위해 vector 라이브러리를 사용하였습니다.




 <b>국민대학교</b> <b>소프트웨어학부</b> <b>C++ 프로젝트</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Snake game	
	<b>팀 명</b>	<b>HelloSnake</b>	
	Confidential Restricted	Version 2.0	2020-JUN-14

## 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.

1. Map 을 window 위에 표현한 뒤에 snake 를 출력하는 방법이 어려웠으나 map 생성시 백터로 좌표를 생성하여 지정하였습니다.
2. -Incursesw 를 사용하면, snake 의 몸이 갑자기 안 따라오던 지 하는 등의 오류가 있었습니다. -Incurses 로 고치면 실행은 잘 되지만 폰트가 깨져서 나타나는 현상이 있어서 수정하지 않았습니다.
3. 아이템 생성하는 부분이 어려웠는데, map 생성 때 좌표 같이 생성하는 걸로 해결하였습니다.
4. Snake 가 Gate 를 통과하여 다음 Gate 로 나올 때의 방향 설정을 해주는 부분이 어려웠었습니다. 출구 Gate 에서 벽이 있는지를 검사하도록 해주고, 이동하던 방향에 벽이 있으면 다음 오른쪽 방향으로 넘어가 검사해주도록 하여 해결하였습니다.
5. 미션 항목을 체크하는 것이 어려웠습니다. 백터를 사용해야 하나 고민하였으나 각 아이템이나 게이트 형성 부분에서 count 변수를 만드는 것으로 해결하였습니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>C++ 프로젝트</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Snake game	
	<b>팀 명</b>	<b>HelloSnake</b>	
	Confidential Restricted	Version 2.0	2020-JUN-14

## 2.2.5 결과물 목록

작성요령 (5 점)

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

**-map.h**

map 의 정보를 담아 놓은 헤더 파일이다. 2 차원 배열 텍스트로 이루어져 있다. Level 에 따라 5 단계의 map 으로 구성되어 있다. Map 을 감싸는 가장 끝 벽은 3, snake 가 접촉할 수 없는 모서리 부분은 2, 일반 벽은 1 로 이루어져 있습니다. 0 으로 이루어진 부분은 snake 가 움직일 수 있는 공간입니다.

**-snake.cpp**

Class vector, class snake, 게임을 돌리는 데에 직접적으로 필요한 몇 가지 함수들과 게임이 구현된 파일입니다. 다른 라이브러리 또한 여기에 including 되어있습니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>C++ 프로젝트</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Snake game	
	<b>팀 명</b>	HelloSnake	
	Confidential Restricted	Version 2.0	2020-JUN-14

### 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.

#### 민경서

처음에는 C++ 언어로 스네이크 게임을 만드는 것에 많은 두려움이 있었습니다. 혼자서는 할 수 없을 것 같아 다른 사람과 함께 같이 해야겠다고 생각했습니다. 작년에도 비대면이여서 아는 동기가 없었기 때문에 같이할 사람을 구하는 것에서부터 난관이 생겼었습니다. 같이할 사람을 구하기 위해 한 동기에게 연락을 드려 같이 하자고 부탁드렸더니 좋다고 하여 같이 하게 되었습니다. 저희는 먼 곳에 살지만 같이 한 번은 만나서 계획을 짜고 싶어 첫 만남에 어떻게 스네이크 게임을 만들어 나갈지 많은 이야기를 했습니다. 저희는 보통 실시간 코딩을 통해 함께 코드를 짰고 map 을 어떤 식으로 구성할지 함께 논의한 후, 조원이 만들어준 맵을 바탕으로 growth, poison item 을 출현시키고 gate 구현을 하였습니다. 특히 gate 를 구현하는 과정에서 많은 어려움을 느꼈는데 조원과 함께 실시간 코딩을 하면서 밤을 새우며 에러의 원인을 찾고 해결해 나갔습니다. 막혔을 때마다 포기하고 싶은 마음이 들었지만 어느 순간 해결되면 그 희열을 잊을 수가 없어 끝까지 완성할 수 있었습니다. 저희는 코드를 처음부터 다시 짜는 것을 반복하기도 했습니다. 처음에는 헤더 파일과 cpp 파일을 나누지 않고도 코드를 짜보기도 했고, 다시 분리해서 짜보기도 했고 많은 시행착오를 거쳤습니다. 시간이 굉장히 오래 걸려 조원에게도 미안했었고 둘 다 많이 힘들어 했습니다. 비대면 상황이었기 때문에 서로의 일정을 맞추는 것도 굉장히 어려웠습니다. 하지만 프로젝트를 끝내면서 많은 성장을 했다는 것을 느꼈습니다. 예를 들어, 알고리즘을 생각해내도 그 생각을 코드로 옮기는 것은 어려운데 이번 프로젝트를 통해서 조금이나마 성장할 수 있었던 것 같습니다. 조원이 있었기에 프로젝트를 완성할 수 있었고 이 기회를 제공해준 C++수업이 저에게 많은 도움이 되었습니다.

#### 김규리

저는 1, 2, 5 단계를 맡아 구현하였습니다. 로직을 잘 짜는 것도 아니면서, 프로젝트를 하게 되어 많이 두려웠습니다. Ncurses 를 사용하려면 따로 공부를 해야 했고, 팀을 짜야만 했습니다. 민폐가 될까 두렵고 코로나로 인해 동기를 전혀 알지 못한 상황이었는데, 다행히 먼저 연락을 주어 같이 하게 되었습니다. clear 로 지웠다가 나타나게 하려했으나 제대로 사용하지 못하는 등 아쉬움이 많지만, 그래도 많은 디버깅 끝에 Window 와 snake 를 출력하게 되었습니다. 그리고 snake 가 루프를 썼음에도 불구하고 누를 때만 움직여서 아무것도 입력 받지 않을 경우까지 switch 에 넣어 놓는 등 많은 경우를 거치게 되었습니다. 갑자기 snake 가 움직이지 않는 등 다양한 버그들이 나타났지만, 많은 과정을 겪고 간신히 완성되었습니다. 조원 서로 github 와 구글시트, 카톡, zoom 을 이용해

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>C++ 프로젝트</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Snake game	
	<b>팀 명</b>	<b>HelloSnake</b>	
	Confidential Restricted	Version 2.0	2020-JUN-14

실시간으로 화면을 공유하는 등, 정보를 주고받으며 프로젝트에 열심히 임하였습니다. 이런 과정들이 스스로에게 가장 큰 도움이 된 것 같습니다. 그리고 ncurses 라는 새로운 라이브러리를 배우면서 생각보다 구현할 수 있는 방법이 무궁무진하다는 것을 알게 되었습니다. C++라는 새로운 언어로 프로젝트를 진행하면서 어느정도 익숙해지는 과정을 겪었다고 생각합니다. 긴 시간을 투자하면서 쉬고 싶다는 생각이 절로 났지만 굉장히 의미가 있는 시간이었습니다.

## 4 참고 문헌

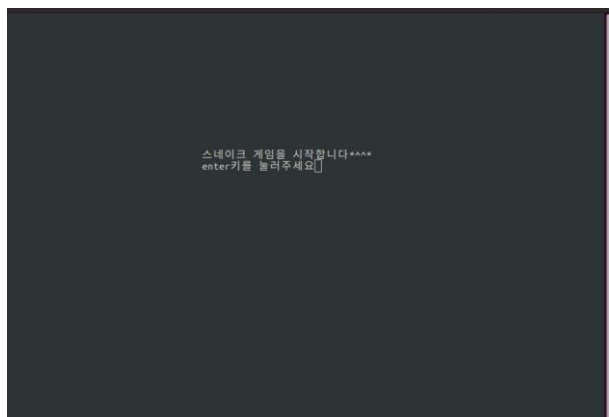
번호	종류	제목	출처	발행년도	저자	기타
1	웹사이트	<time>	<a href="https://hijuworld.tistory.com/1">https://hijuworld.tistory.com/1</a>	2018	공도리	
2	웹사이트	<ncurses 명령어>	<a href="http://comedudb6.knue.ac.kr/tykim/Myhome/Cnote/chap4-CursesLecture.pdf">http://comedudb6.knue.ac.kr/tykim/Myhome/Cnote/chap4-CursesLecture.pdf</a>			
3	웹사이트	<operator overlading>	<a href="https://0-sunny.tistory.com/56">https://0-sunny.tistory.com/56</a>	2020	영년	

## 5 부록


### 작성요령 (15 점)

프로젝트의 결과물을 사용하기 위한 방법에 대해서 작성하세요.

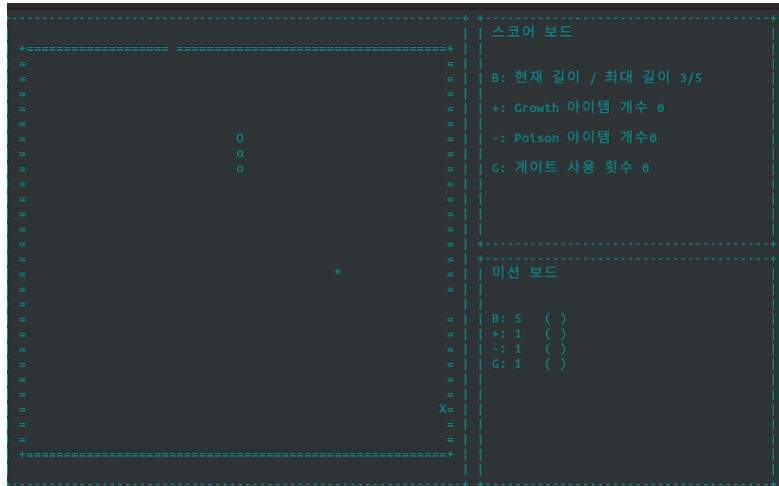
### 5.1 사용자 매뉴얼



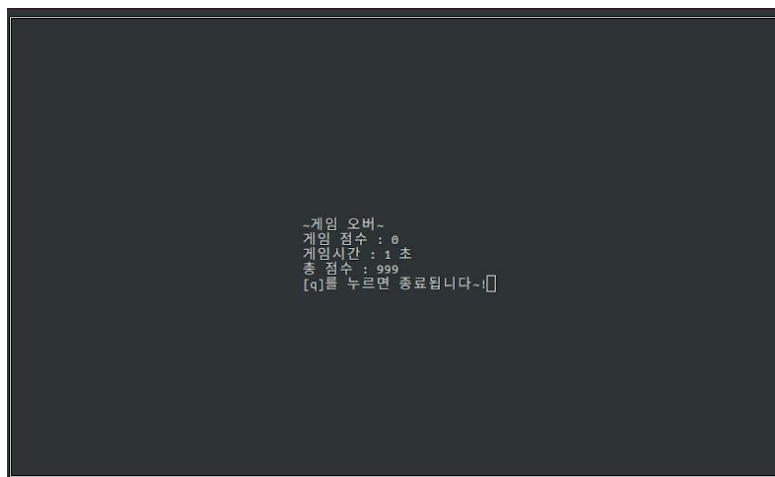
#### 1. enter 키를 눌러주세요

 <div> <b>국민대학교</b>  <b>소프트웨어학부</b>  <b>C++ 프로젝트</b> </div>	결과보고서		
	프로젝트 명	Snake game	
	팀 명	HelloSnake	
	Confidential Restricted	Version 2.0	2020-JUN-14

2. snake 와 함께 map 이 뜨며 바로 시작하게 됩니다.



3. 방향키를 누르면 snake 가 움직이게 됩니다.  
 4. 게임이 끝났다면 이 장면이 뜨게 됩니다.



## 5.2 설치 방법

프로그램 컴파일 및 실행 방법 포함

1. map.h, snake.cpp 파일을 다운 받습니다.
2. 다운받은 파일을 열고, (linux 우분투 기준)

`[g++ -o snake snake.cpp -lncursesw]`

의 명령어로 컴파일을 해준 뒤, [ ./snake]등의 방법으로 실행해주면 됩니다.