**C++프로그래밍**

**프로젝트**

| **프로젝트 명** | *Snake Game* |
| --- | --- |
| **팀 명** | *목신조 (13조)* |
| **문서 제목** | 결과보고서 |

| **Version** | **1.3** |
| --- | --- |
| **Date** | 2023-Jun-17 |

| **팀원** | 목진협 (팀장) |
| --- | --- |
| 신진욱 |
| 조민형 |
|  |
|  |

| **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 “목신조(13조)”의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 “목신조(13조)”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다. |
| --- |

**문서 정보 / 수정 내역**

| **Filename** | 최종보고서-SnakeGame.doc |
| --- | --- |
| **원안작성자** | 목진협, 신진욱, 조민형 |
| **수정작업자** | 목진협, 신진욱, 조민형 |

| 수정날짜 | 대표수정자 | Revision | 추가/수정 항목 | 내 용 |
| --- | --- | --- | --- | --- |
| 2023-06-15 | 목진협 | 1.0 | 최초 작성 |  |
| 2023-06-16 | 신진욱 | 1.1 | 내용 수정 | 보고서 내용 추가 |
| 2023-06-17 | 조민형 | 1.2 | 내용 추가 | 향후 추진 계획 추가 |
| 2023-06-17 | 목진협 | 1.3 | 내용 수정 | 오타 제거 및 형식 정리 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**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](#_1t3h5sf)  해결 방안 6

[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)

## **1** **개요**

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

**프로젝트의 전체적인 구조 및 개발 내용을 명확하게 기술한다.**

## **1.1** **프로젝트 소개**

국민대학교 C++프로그래밍 교과목에서 C++언어와 ncurses 라이브러리를 활용하여 제작한 Snake Game 프로젝트입니다.

## **1.2** **개발 방법**

- 개발OS : Linux Ubuntu 16.04 LTS 기준 ( + Mac OS, Window )

- 언어: C++ 14

## **1.3** **사용 라이브러리**

- ncursesw 라이브러리

## **1.4** **설치 방법(Linux기준)**

- sudo apt-get update

- sudo apt-get install libcurses5-dev libcursesw5-dev

# **2 개발 내용 및 결과물**

## **2.1 목표**

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

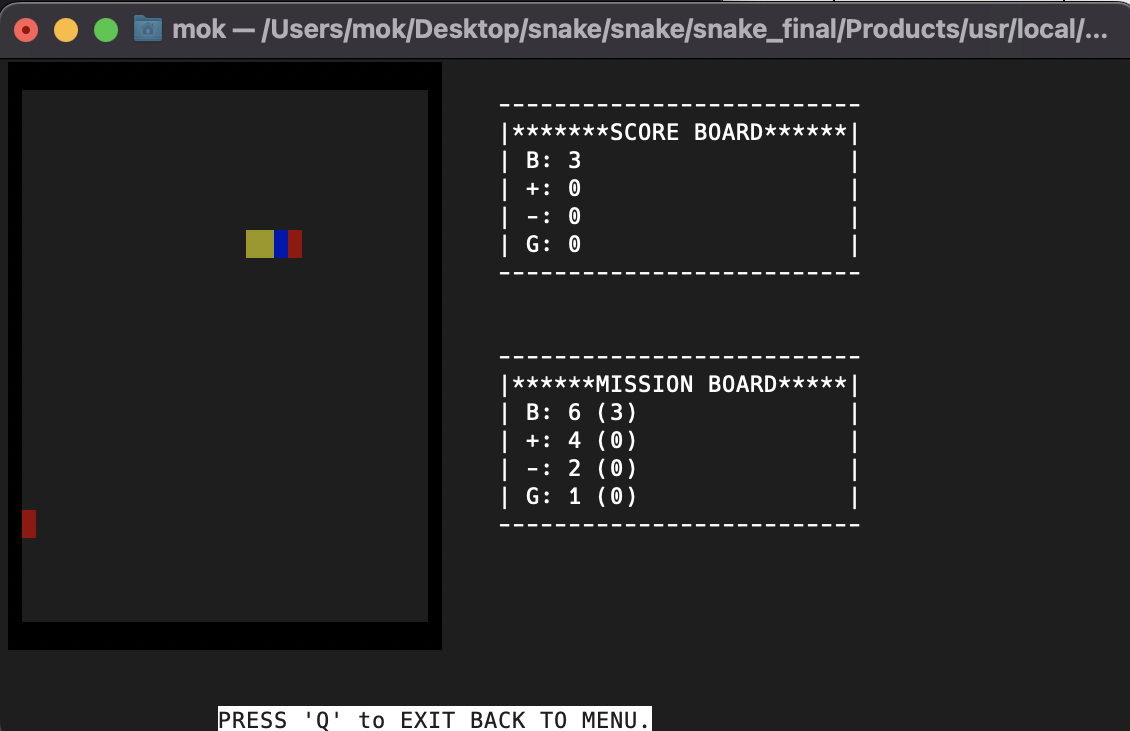
| 적용단계 | 내용 | 적용 여부 |
| --- | --- | --- |
| 1단계 | Map의 구현 | 적용 |
| 2단계 | Snake 표현 및 조작 | 적용 |
| 3단계 | Item 요소의 구현 | 적용 |
| 4단계 | Gate 요소의 구현 | 적용 |
| 5단계 | 점수 요소의 구현 | 적용 |

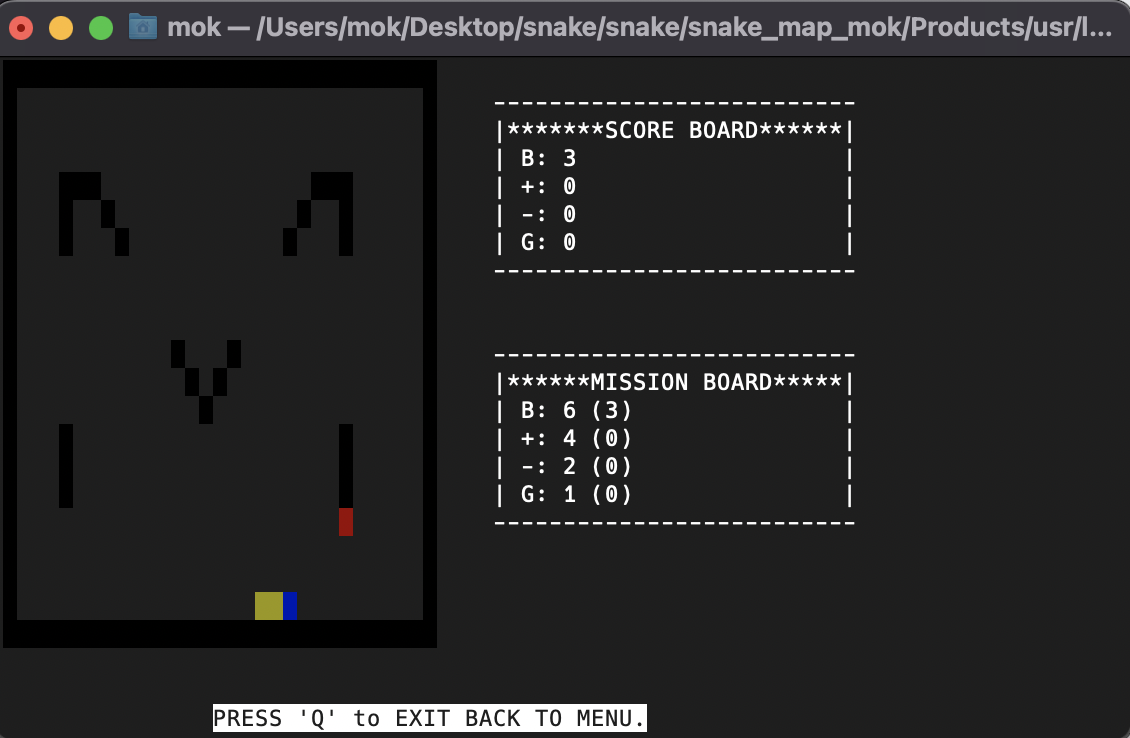
**프로젝트의 각 적용 단계별 구현 목표를 명확하게 제시한다. 제시한 권고안의 내용을 포함하여, 변경된 부분 등을 구체적으로 단계별 구현 목표를 작성해야 한다.**

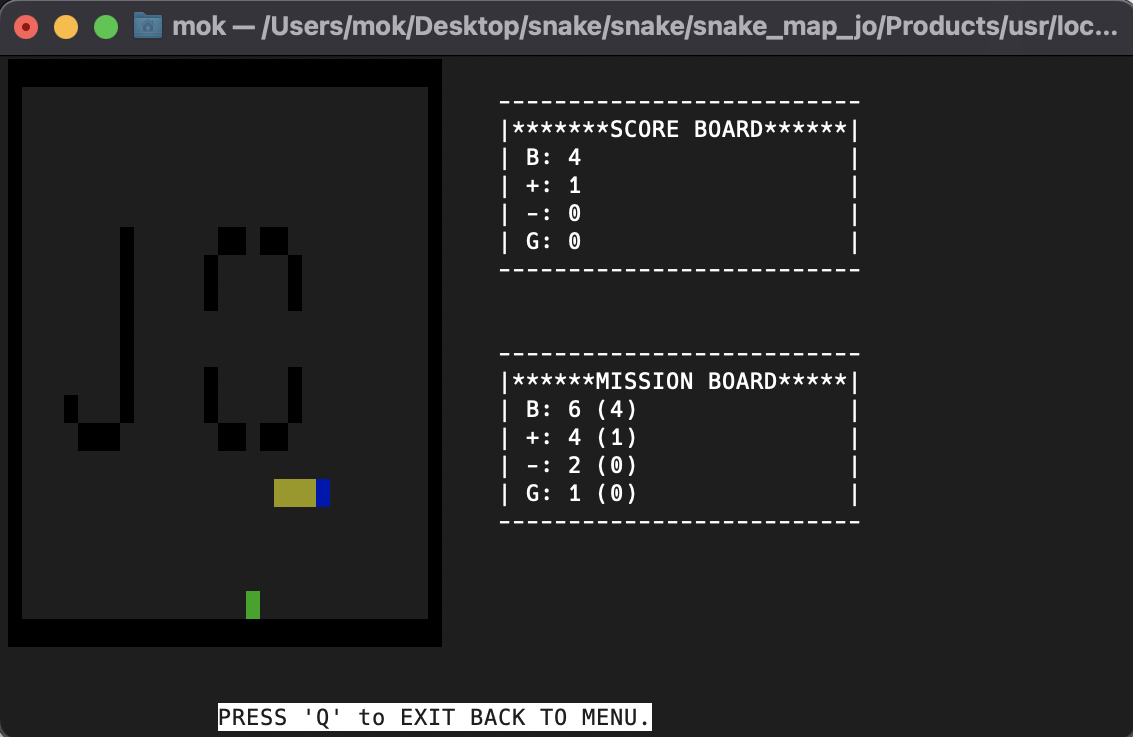
**1. Map**

· 각 스테이지에 따른 Map을 구현한다.

1. 스테이지1 : MAP1 - Basic



2. 스테이지2 : MAP2 - M (Mok 목진협)

3. 스테이지3 : MAP3 - Jo (Jo 조민형)

4. 스테이지4 : MAP4 - S (Shin 신진욱)

**2. Snake**

· 시작 길이: 3

· 시작 진행 방향: 오른쪽

· 머리의 색은 파랑, 몸통의 색은 노란색이다.

· 사용자가 방향키를 입력하면, Snake의 좌표는 1초마다 진행 방향으로 변한다.

· 몸의 길이가 3보다 작아지면 게임이 종료된다.

· 머리가 벽이나 몸통에 닿으면 게임이 종료된다.

· 머리와 아이템의 좌표가 같아질 때, 아이템을 얻는다.

**3. Item**

· 아이템은 5초마다 랜덤하게 생성.

· 아이템은 Snake, Wall과 겹치지 않도록 생성한다.

· 최소 1개부터 최대 3개까지의 아이템이 랜덤하게 생성된다.

· 아이템은 Growth와 Poison가 랜덤하게 생성된다.

- Growth: 몸 길이 +1

- Poison: 몸 길이 -1

· 아이템은 획득 시 Map에서 사라진다.

**4. Gate**

· Gate는 Snake 길이가 6 이상일 때, 생성된다.

· 10초마다 Gate가 새로 생성되는데, Snake가 통과 중일 때는 생성되지 않는다.

· Gate는 IMMUNE WALL에 생성되지 않는다.

**5. Score**

Score Board에서는 현재까지 Snake가 획득한 점수들을 출력한다.

· B: Snake의 현재 길이

· +: 현재까지 획득한 Growth 아이템 수

· -: 현재까지 획득한 Poison 아이템 수

· G: 현재까지 통과한 Gate 수

## **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.** |
| --- |

**1. Map**

· Map 크기: 21 X 31

· Map은 Stage마다 다르게 구성.

· Map은 3차원의 구조로 되어있다. [STAGE][ROW][COL]

· 각 Cell은 WALL/IMMUNEWALL 에 해당하는 값이다.

- MAP[STAGE][ROW][COL] == 0; -> WALL

- MAP[STAGE][ROW][COL] == 2; -> IMMUNEWALL

**2. Snake**

**2.1 Snake Class**

· Snake Class에서는 Snake 생성, 아이템 생성, 초기 Map구성, Gate출현의 기능을 수행하는 메소드가 존재한다.

**2.2 Snake 생성**

· Snake클래스의 생성자를 통해 Snake를 생성한다. 인자 값으로는 stage의 값을 넘겨받게 된다.

· Snake는 deque 구조로 생성되며 배열의 원소는 Map의 각 좌표들이다.

**2.3 방향키 입력**

· 사용자가 방향키를 입력하게 되면 Snake가 방향키에 따라 움직인다.

· 다음과 같은 상황에는 방향키 입력이 무시된다.

- 현재 진행방향과 같은 방향

· 다음과 같은 상황에는 게임 오버된다.

- 현재 진행방향과 반대 방향

**2.4 Snake의 움직임 표현**

· 방향키 입력에 따른 좌표 변화

| **방향키** | **좌표변화** |
| --- | --- |
| **상** | 행이 1 만큼 감소 |
| **하** | 행이 1 만큼 증가 |
| **좌** | 열이 1 만큼 감소 |
| **우** | 열이 1 만큼 증가 |

· 구현방법

- 변화된 좌표를 snake(deque) front에 push한다.

- pop하여 맨 뒤 원소를 삭제한다.

**2.5 화면 출력**

· Snake(deque)의 위치를 표현하기 위해 화면을 매초 refresh하여 출력해준다.

**3. Item**

**3.1 Item 출현**

· Item 출현 시 종류를 랜덤하게 한다.

· function rand\_score(): +1 또는 -1을 리턴한다.

- (+1): Growth 아이템

- (-1): Poison 아이템

**3.2 Item 획득**

· Snake Head와 Item이 부딪히게 됐을 경우 아이템을 획득한다.

· Snake Head: cells(snake의 deque).front()

· Head와 Item의 좌표가 같아질 경우, 획득한 아이템에 대한 정보들을 저장한다.

- 현재 아이템의 Iterator => 아이템을 먹고난 후의 삭제

- 현재 아이템의 좌표

- 아이템 플래그 => Poison과 Growth를 구별

· Item 획득 시, 해당 Item은 화면에서 사라진다.

- items(MAP에 나타나는 아이템 배열).erase(curitem.t);

· Item 획득 시, 현재까지 획득한 Item 개수를 저장하는 변수를 1 증가시킨다.

- Score Board에서 Growth, Poison을 표시.

**4. Gate**

**4.1 Gate 출현**

· Snake의 길이가 6 이상일 때 출현한다.

· 10초마다 새로운 Gate를 생성한다.

· Snake가 Gate를 통과 중인지 체크하고 통과 중이면 생성하지 않는다.

· Wall중에(Not immune) Gate 한 쌍을 랜덤하게 출현시킨다.

- Snake/Item의 좌표와 겹치지 않게 생성한다.

- Gate는 한 쌍이므로, 총 두개의 Gate를 생성하여 deque에 저장한다.

**4.2 Gate 통과**

· Gate를 통과 중인지 체크하고 통과 중이라면 flag값을 true로 설정한다.

· 매번 출구 좌표에 Gate의 양쪽 값을 대입하다가 Gate 통과 flag가 true일 시

해당 게이트 좌표의 대입을 건너뛴다.

=> 출구 좌표는 통과 중이지 않은 좌표로 설정된다.

· Gate flag가 true일 경우,

- 출구 Gate의 좌표에 따라 이동방향을 설정해준다.

| **Gate(out)** | **Direction** |
| --- | --- |
| Row == 0 | ↓ |
| Row == end | ↑ |
| Col == 0 | → |
| Col == end | ← |

- Gate통과 횟수를 저장하는 변수를 1 증가시킨다.(Score Board)

**5. Score & Mission**

**5.1 Score Board**

· Score Board에서는 현재까지 Snake가 획득한 점수들을 출력한다.

- Snake.getScore(): Snake의 현재 길이

- Snake.getCntGrowth(): 현재까지 획득한 Growth Item 수

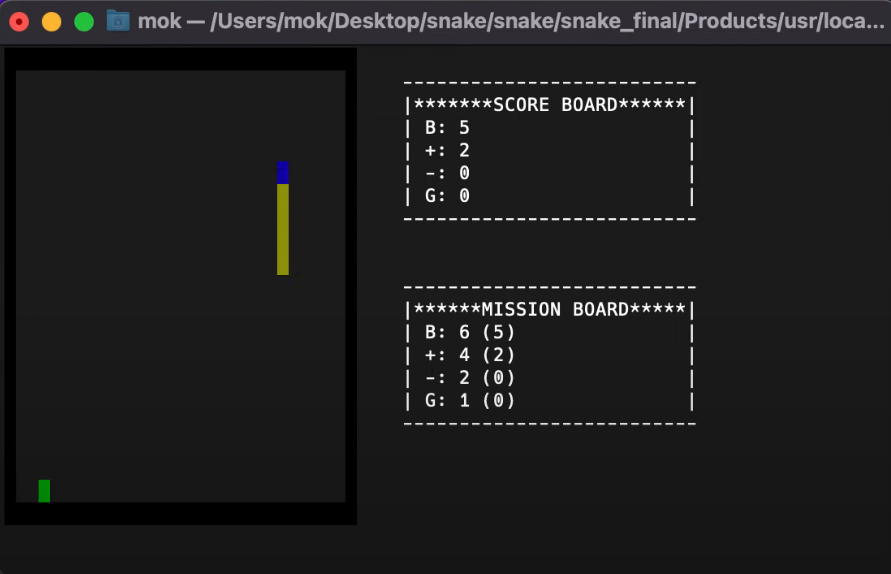
- Snake.getCntPoision(): 현재까지 획득한 Poison Item 수

- Snake.getCntGate(): 현재까지 통과한 Gate 수

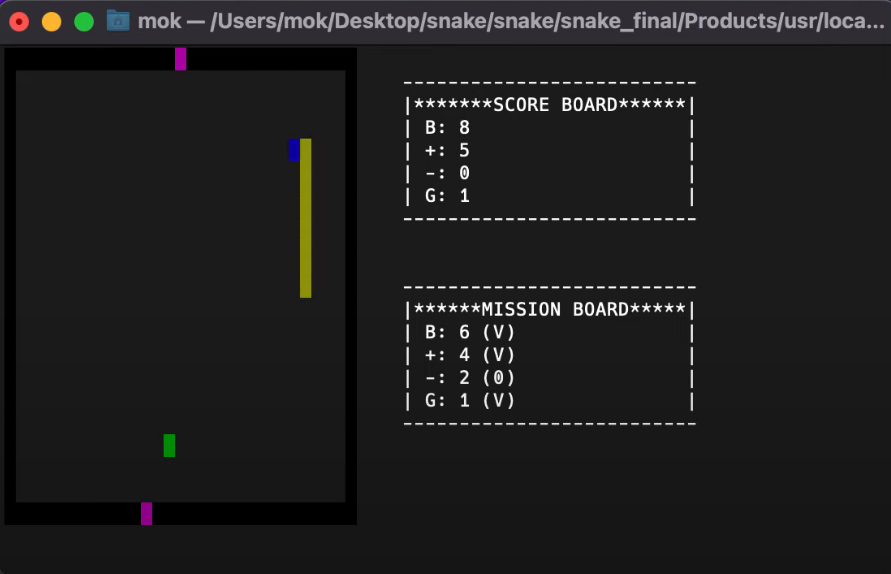
**5.2 Mission Board & Stage**

· Mission Board에서는 각 Stage마다 수행해야 할 Mission들이 출력된다.

Ex)



· 각 Mission을 완수할 경우 괄호 안에 ‘V’자를 출력 한다.



### **2.2.2 시스템 구조 및 설계도**

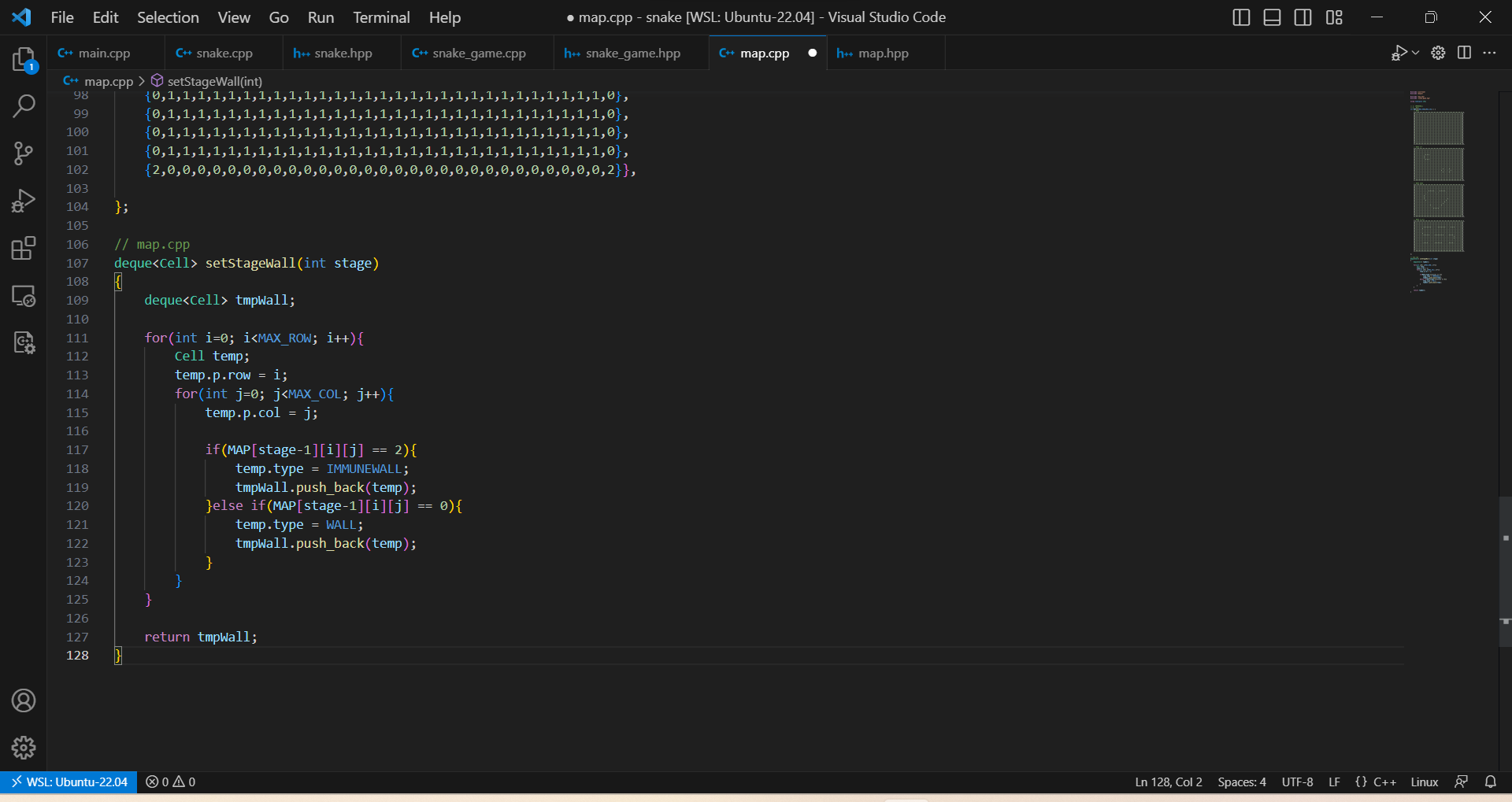
| **작성요령 (30점)**  **프로젝트의 각 세부 목표의 주요 기능(알고리즘 등)에 대해서 기술한다. 세부 목표별로 수정한 프로그램 소스 파일을 나열하고, 해당 파일에서 세부 목표를 달성하기 위해 작성한 클래스/함수에 대해 나열하고, 각 요소에 대해 간략한 설명을 작성한다. 또한 각 요소의 개발자를 명시한다.** |
| --- |

| **적용단계** | **내용** | **소스 파일** |
| --- | --- | --- |
| **1단계** | Map 구현 | map.cpp / snake.cpp |
| **2단계** | Snake 생성 및 조작 | snake.cpp / snake\_game.cpp |
| **3단계** | Item 구현 | snake.cpp / snake\_game.cpp |
| **4단계** | Gate 구현 | map.cpp / snake.cpp |
| **5단계** | 점수 시스템 구현 | snake.cpp / snake\_game.cpp |

**1. Map (개발자 : 신진욱)**

**1.1 setStageWall()**

: 초기 Map을 구성하는 함수



· 임시변수 tmpWall을 deque 자료형으로 선언함.

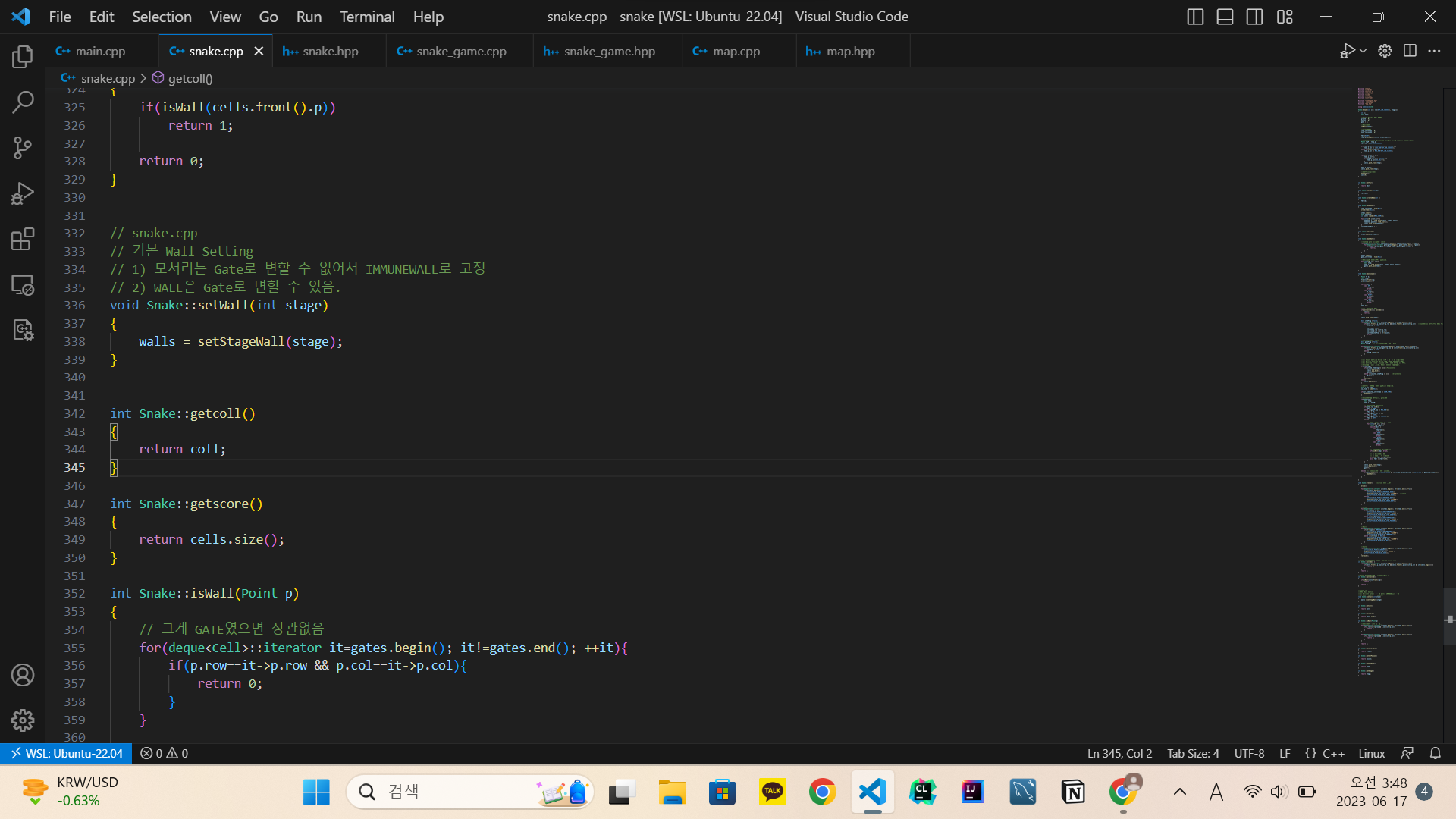
· 미리 저장되어있는 MAP 배열의 값을 확인하여 ImmuneWall/Wall인지를 판단한다.

- MAP[STAGE][ROW][COL]==0; 은 WALL

- MAP[STAGE][ROW][COL]==2; 은 IMMUNEWALL

**1.2 Snake::setWall(int stage)**

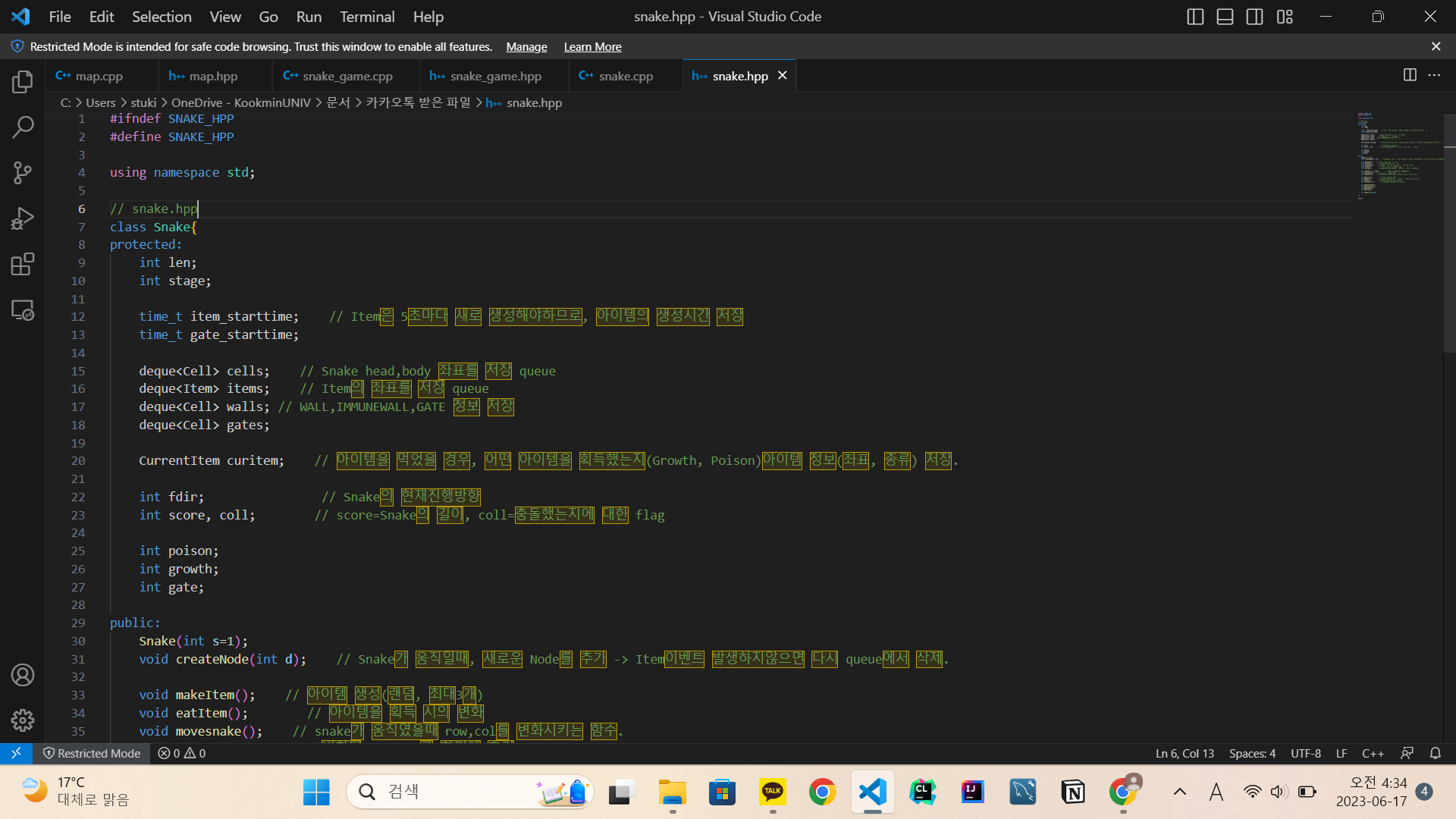
: Snake 클래스의 메소드이며, Snake가 생성됨과 동시에 Map을 생성하도록 해준다. 그리고 setStageWall 메소드를 호출할 때, stage 인자값을 넘겨준다.

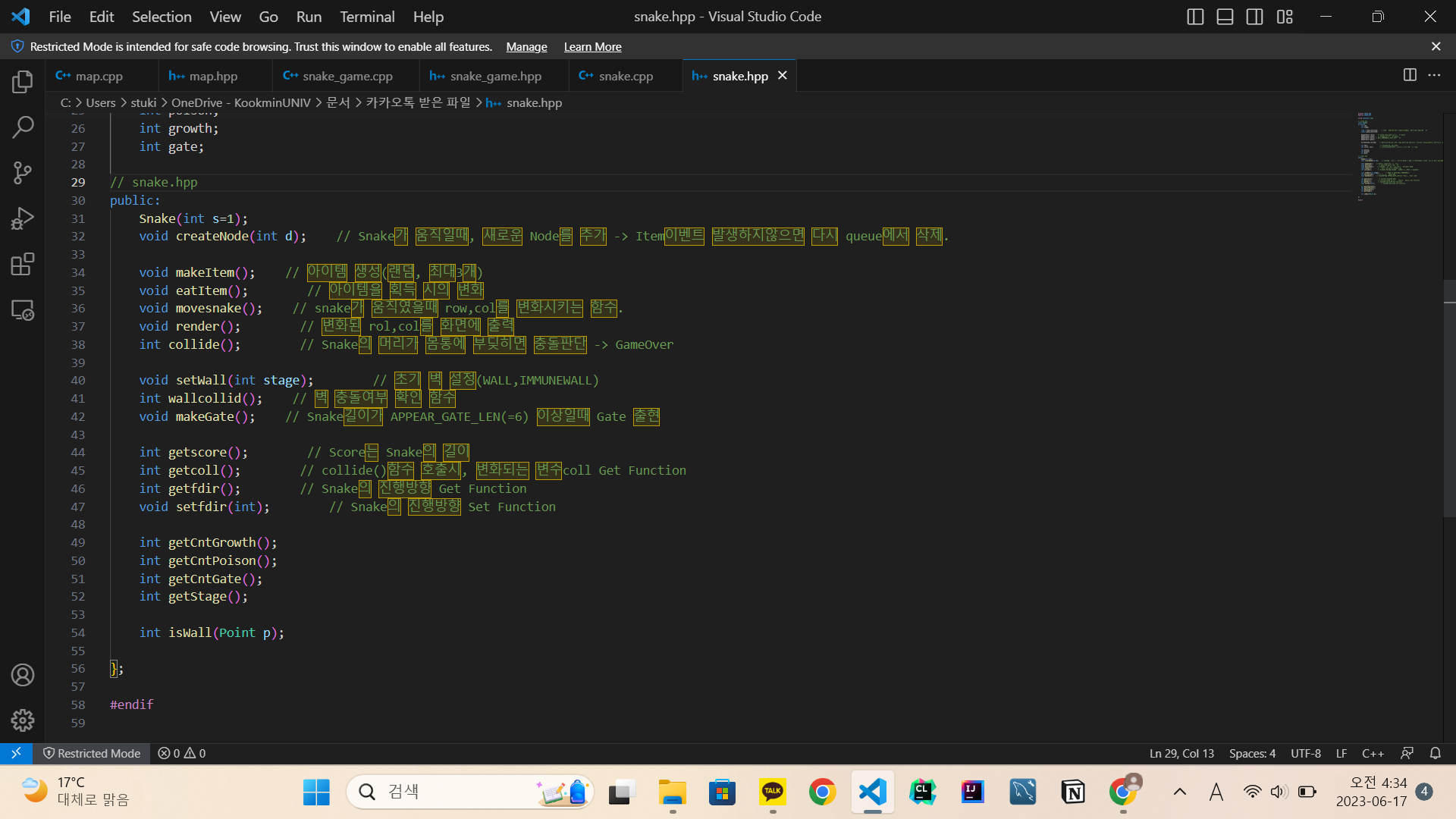


**2. Snake (개발자 : 목진협)**

**2.1 Snake Class**

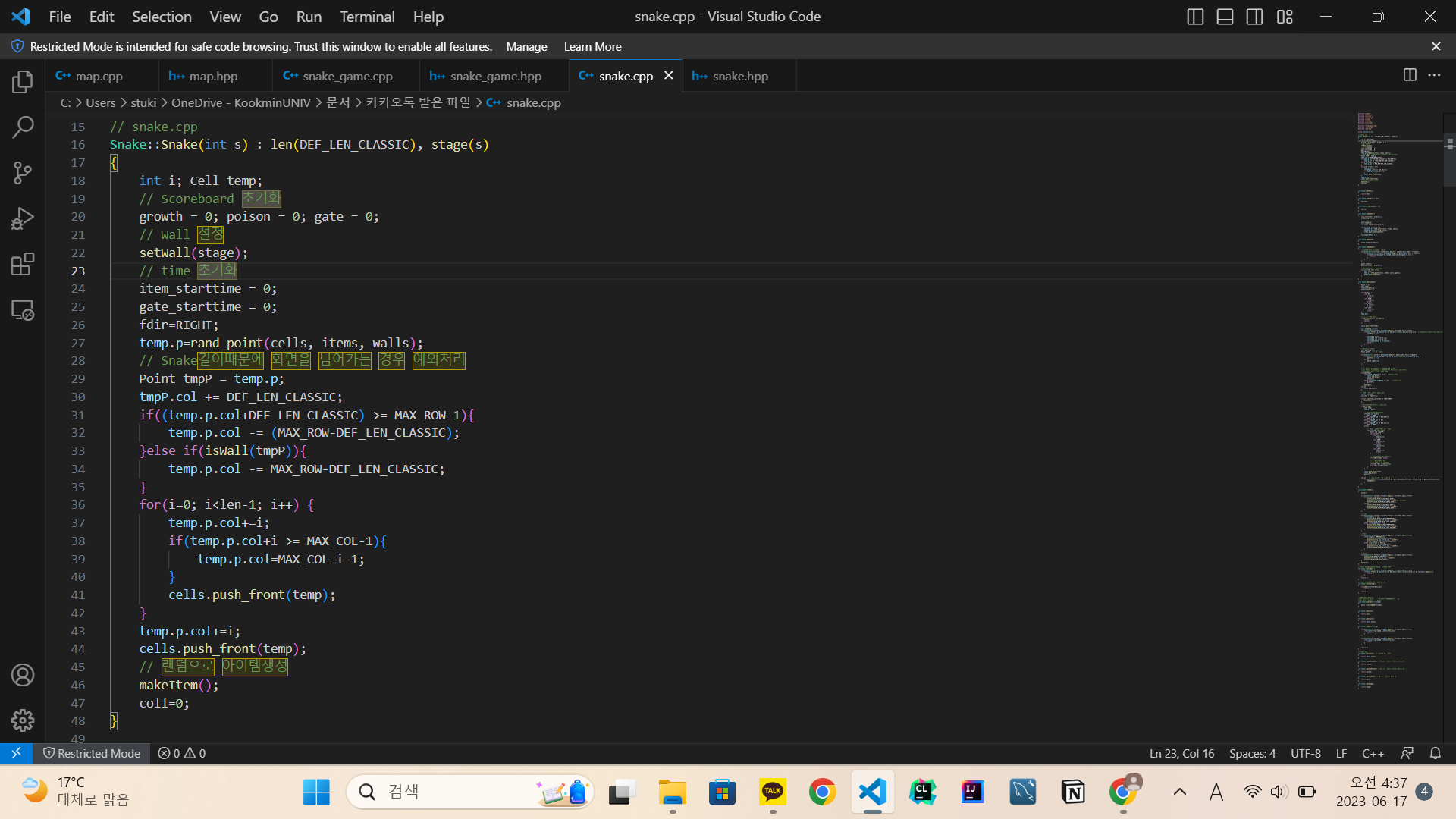
: snake 클래스에서는 시작 맵 구성, snake 생성, 게이트와 아이템 생성을 수행하는 메소드가 존재한다.





**2.2 Snake 생성(Snake 생성자)**

: snake 클래스의 생성자를 통해 snake를 생성한다. 인자값으로는 stage의 값을 넘겨받게 된다.



· 시작 길이가 3인 Snake를 생성한다.

· Score Board에 표시할 점수 변수들을 초기화한다. (growth/poison/gate)

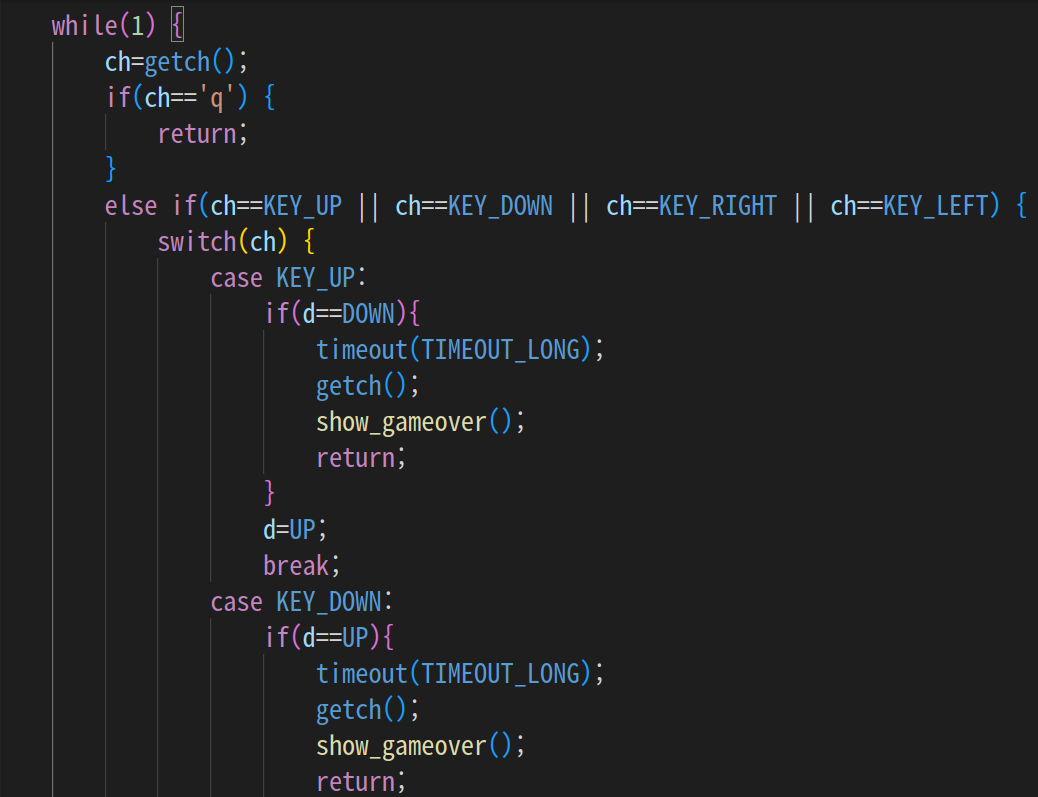
· item 획득한 시간을 초기화한다.

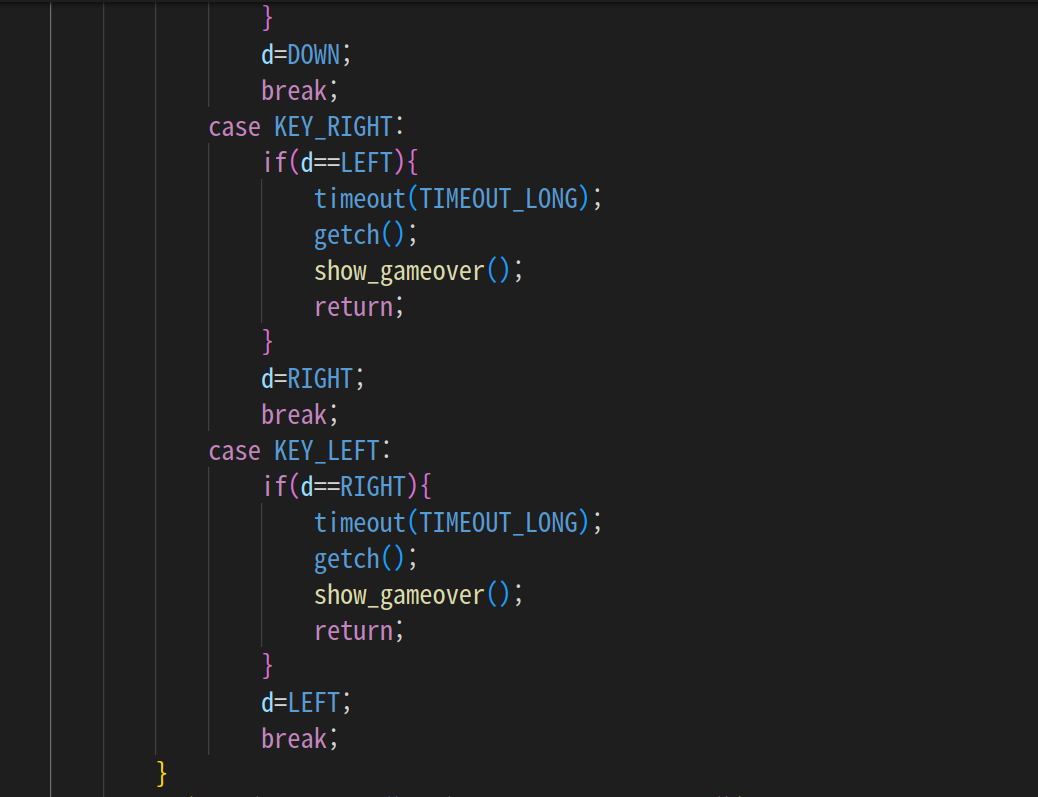
· 시작 이동방향을 오른쪽(fdir=RIGHT)로 설정.

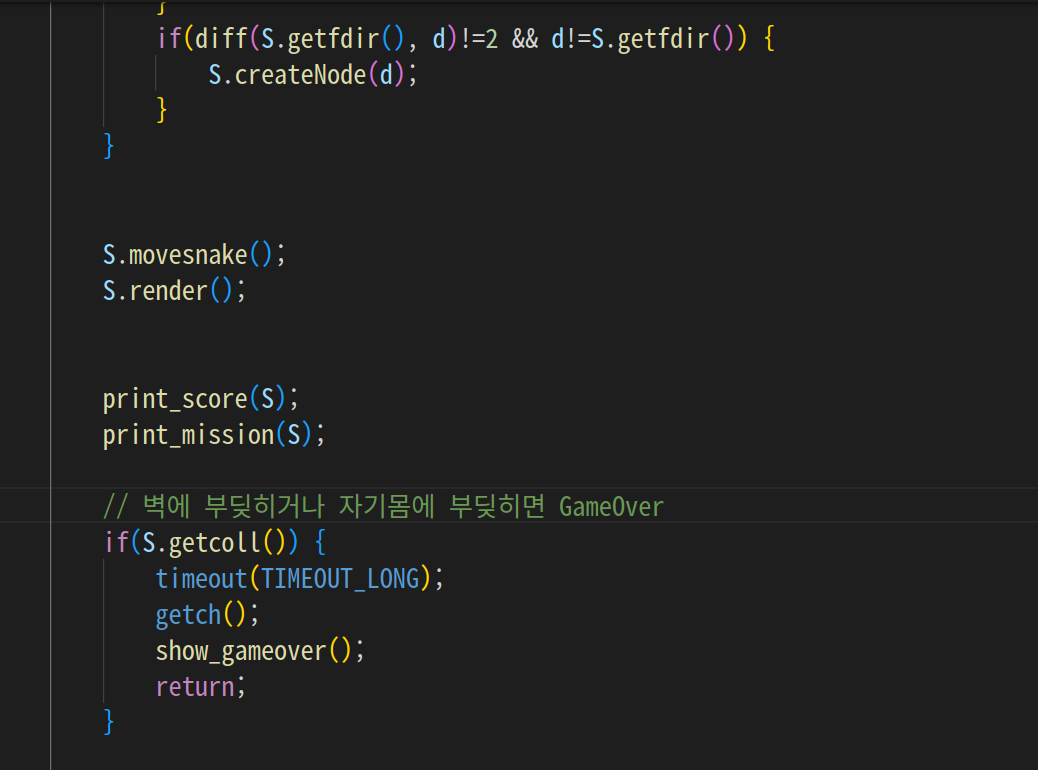
· 초기 Map과 Item을 할당해준다. (좌표는 random하되, 겹치지 않도록 한다.)

**2.3 방향키 입력(게임 진행시)**

: 사용자가 방향키를 입력하면, Snake가 움직인다.









· 다음과 같은 상황에는 방향키 입력이 무시된다.

- 현재 진행방향과 같은 방향

· 다음과 같은 상황에는 게임오버.

- 현재 진행방향과 반대 방향

( 변수 d 에 이전 입력을 저장하여 새로 들어오는 입력이 그 반대 방향일 경우 게임 오버하도록 구현)

· 게임 진행 중, 'q' 입력 시, 메인 화면으로 복귀.

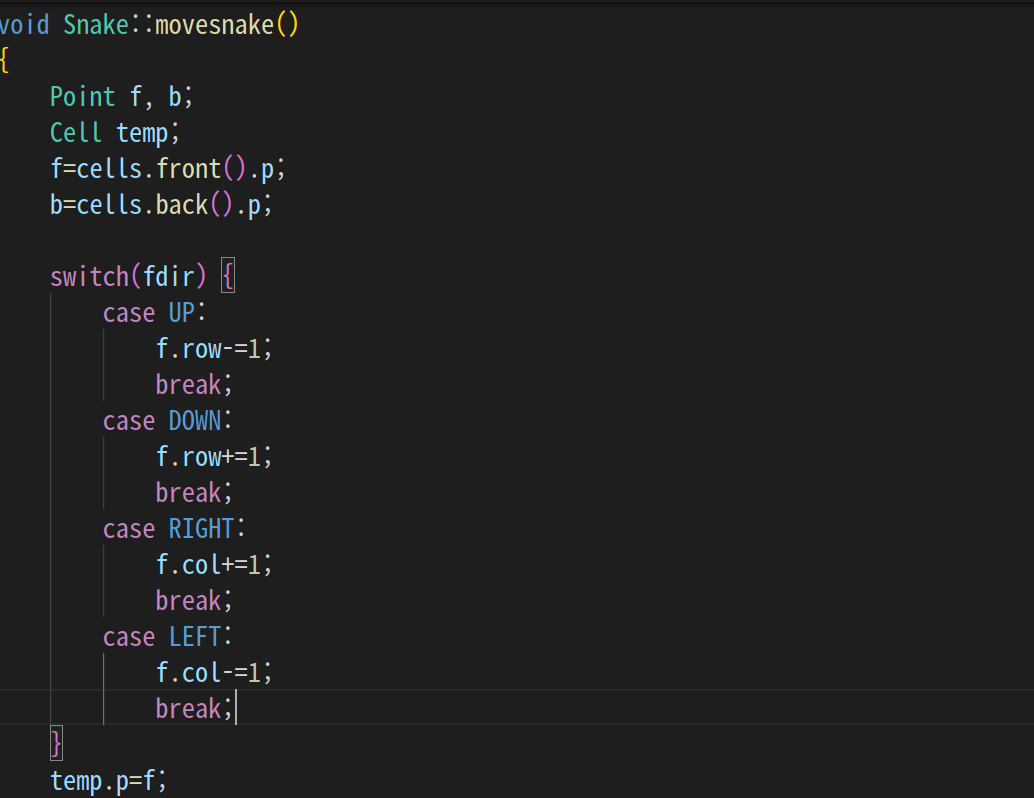
· 벽이나 자기 몸통에 부딪히면 게임 종료

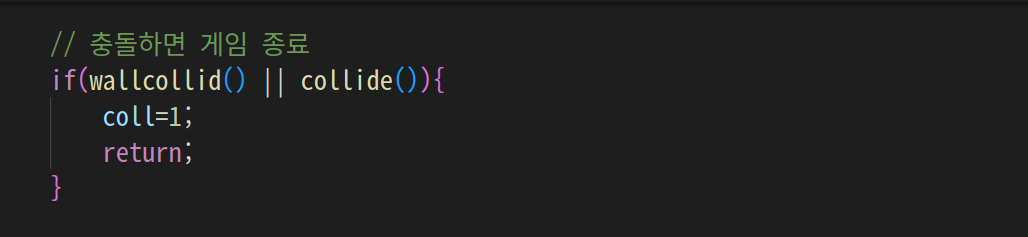
· Snake의 길이가 3보다 작으면 게임 종료

· 해당 Stage의 Mission을 Clear하면, 다음 Stage로 넘어간다.

**2.4 Snake::moveSnake()**

: 사용자가 방향키를 입력하게 되면, Snake의 좌표는 1초마다 진행방향쪽으로 변하게 된다.





· Snake는 기본적으로 “deque” 라는 STL 라이브러리를 사용하여 구현하였다.

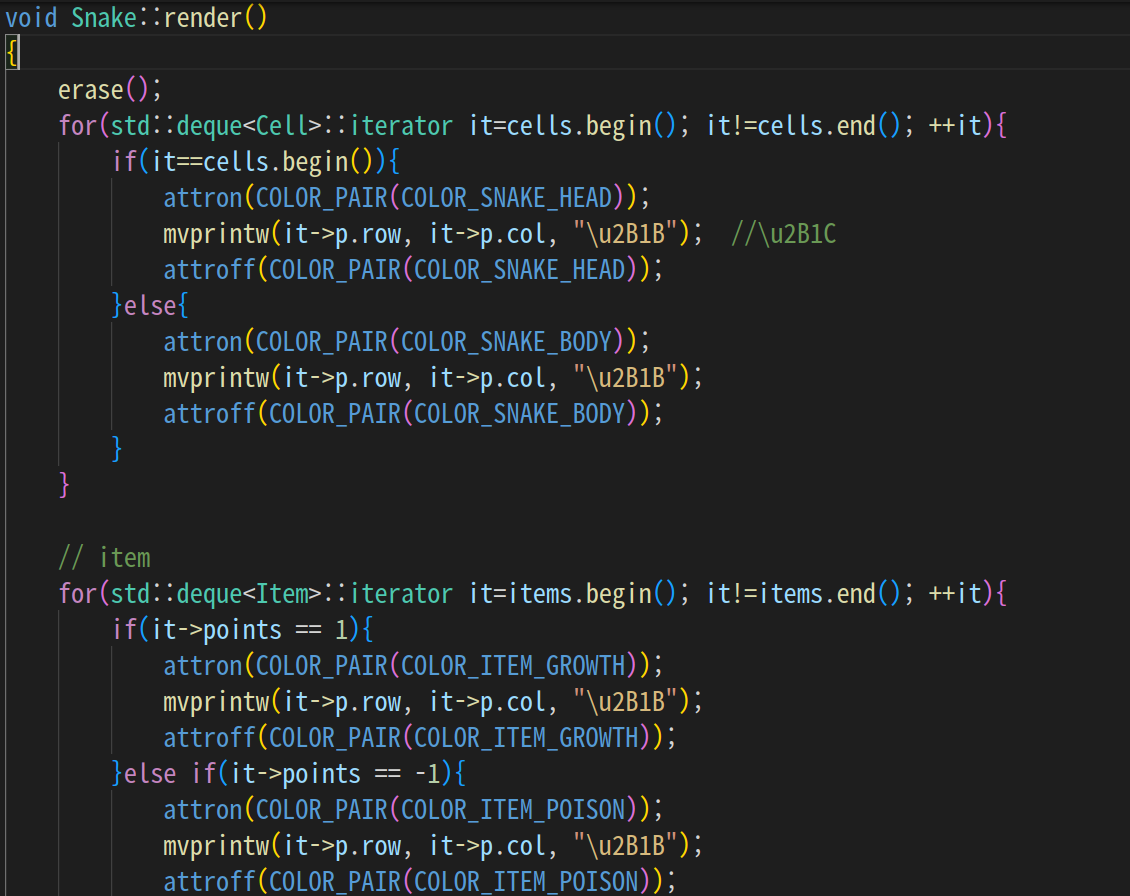
- 변화된 좌표를 push\_front 메소드를 통해 삽입한다.

- 특별한 이벤트가 발생하지 않았을 경우, pop\_back 메소드를 호출하여 맨 뒤 원소를 삭제한다.

· Snake가 Wall에 부딪히거나, 자기 몸에 부딪혔으면 return하여 게임을 종료시킨다.

**2.5 Snake::render()**

: Snake가 움직임을 표현하기위해 매번 refresh하여 출력해준다.

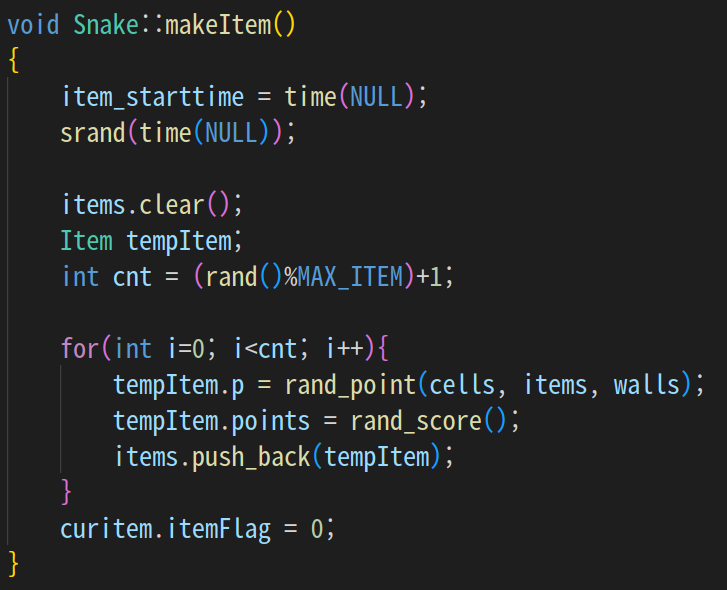




**3. Item (개발자 : 조민형)**

**3.1 Snake::makeItem()**

: 아이템은 5초가 지나면 랜덤하게 나타나게 된다.



· 아이템의 좌표는 Snake와 Wall과 겹치지 않도록 생성한다.

· 최소 1개부터 최대 3개까지의 아이템이 랜덤하게 생성된다.

· 아이템의 종류 또한 랜덤하게 출현한다.

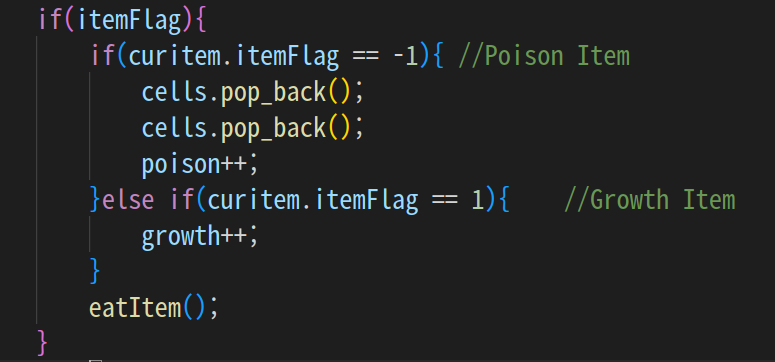
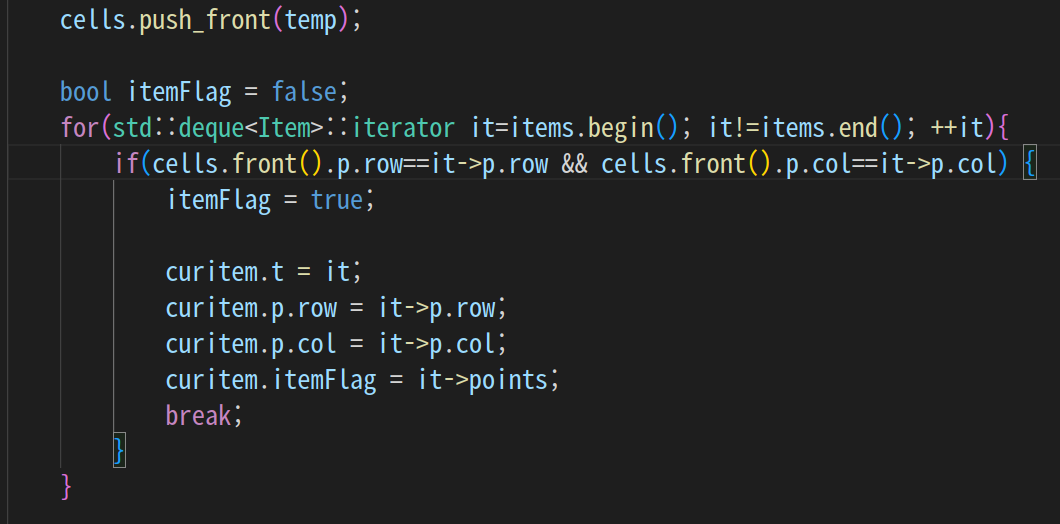
· rand\_score()함수 : 1 또는 -1을 리턴한다.

· Growth아이템 : -1

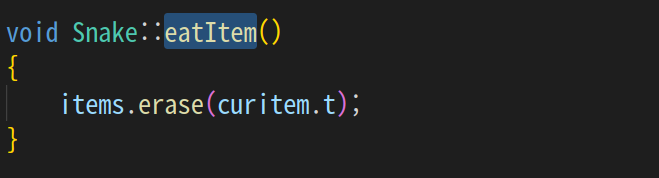
· Poison아이템 : -1

**3.2 아이템 획득**

: Snake Head 부분이 Item과 부딪히게 됐을 경우를 획득했다라고 판단하며, 획득 시 몸의 길이가 증가하거나 감소한다.

****

****

​​****

· Snake Head 부분 = cells.front()

· Head와 아이템 좌표가 같아질 경우, 획득한 아이템에 대한 정보들을 저장한다.

- 현재 아이템의 iterator => 아이템을 먹고난 후의 삭제

- 현재 아이템의 좌표

- 아이템 flag => Poison과 Growth를 구별

· 아이템을 획득하면, 해당 아이템은 화면에서 사라진다.

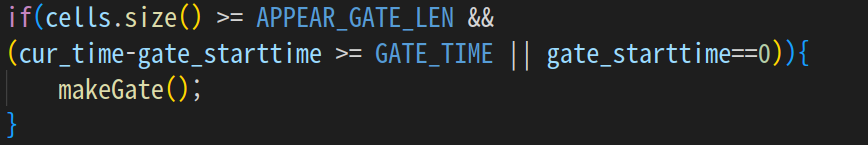
- eatItem()

· 아이템 획득 시, 현재까지 획득한 아이템 갯수를 저장하는 변수를 1 증가시킨다.

**4. Gate (개발자 : 조민형)**

**4.1 Snake::makeGate()**

: Gate는 기본적으로 Snake 길이가 6이상일 때, 출현하게 된다. 10초마다 Gate를 새로 출현하게 되는데, Snake가 통과 중일 때는 생성하지 않는다.



· Snake(cells) 길이가 6 이상이고, gate가 출현한지 10초이상이면 새로운 Gate를 출현시킨다.

· Snake가 gate를 통과 중인지 Check하고, 통과 중이면 생성하지 않는다.

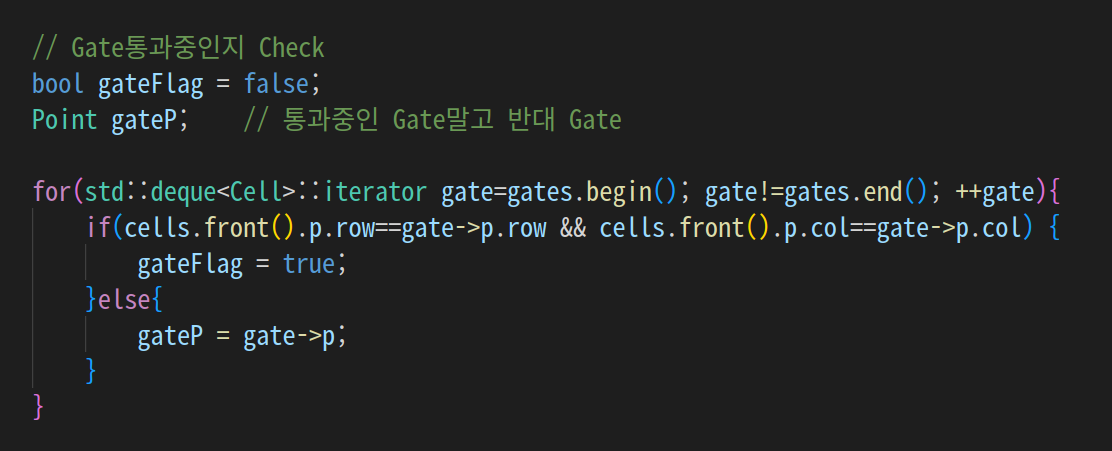
· Wall 중에 Gate 한 쌍을 랜덤하게 생성시킨다.

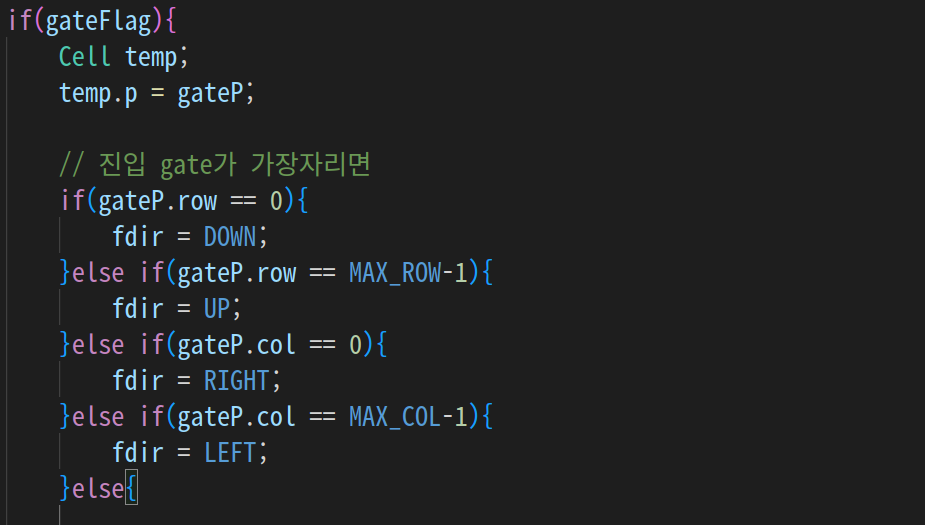
- Snake/Item과 겹치지 않는 좌표를 가져온다.

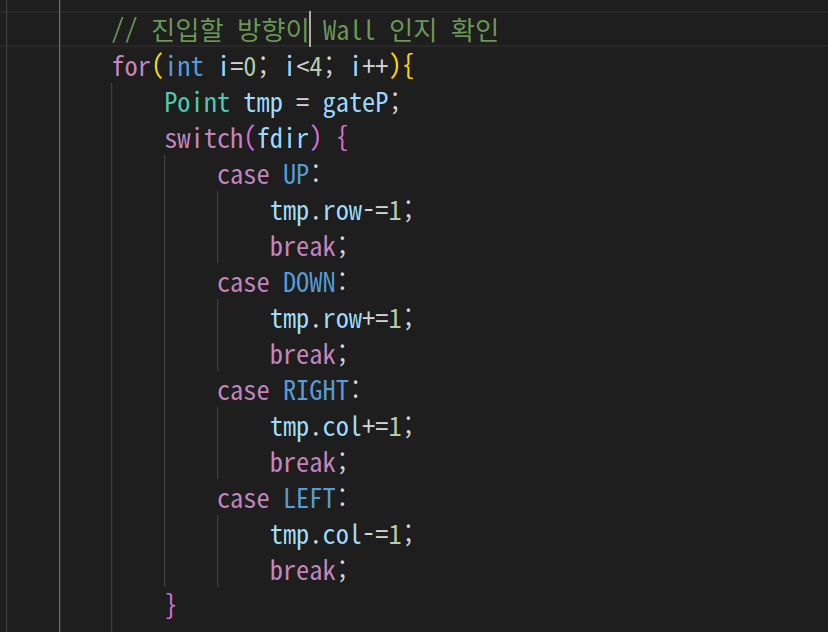
한 쌍의 Gate(2개)를 생성하여 gates deque에 저장한다.

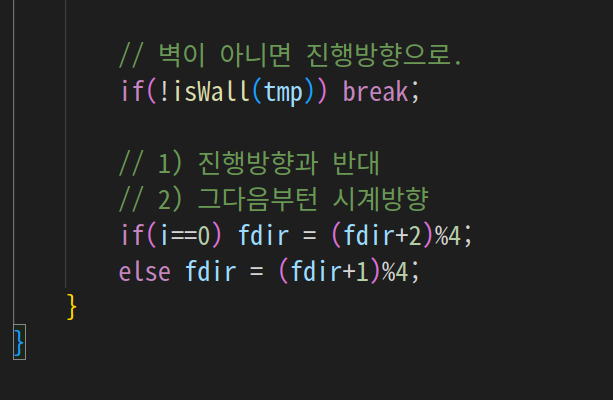
**4.2 Gate 통과**

: Snake가 하나의 Gate에 통과 중인 경우, 다른 Gate로 나온다.









우선 Gate가 통과 중인 지 Check하고, gateFlag 값을 true로 설정한다.

· 그렇지 않은경우, 반대편 Gate의 좌표를 저장한다.

· gateFlag==true일 경우,

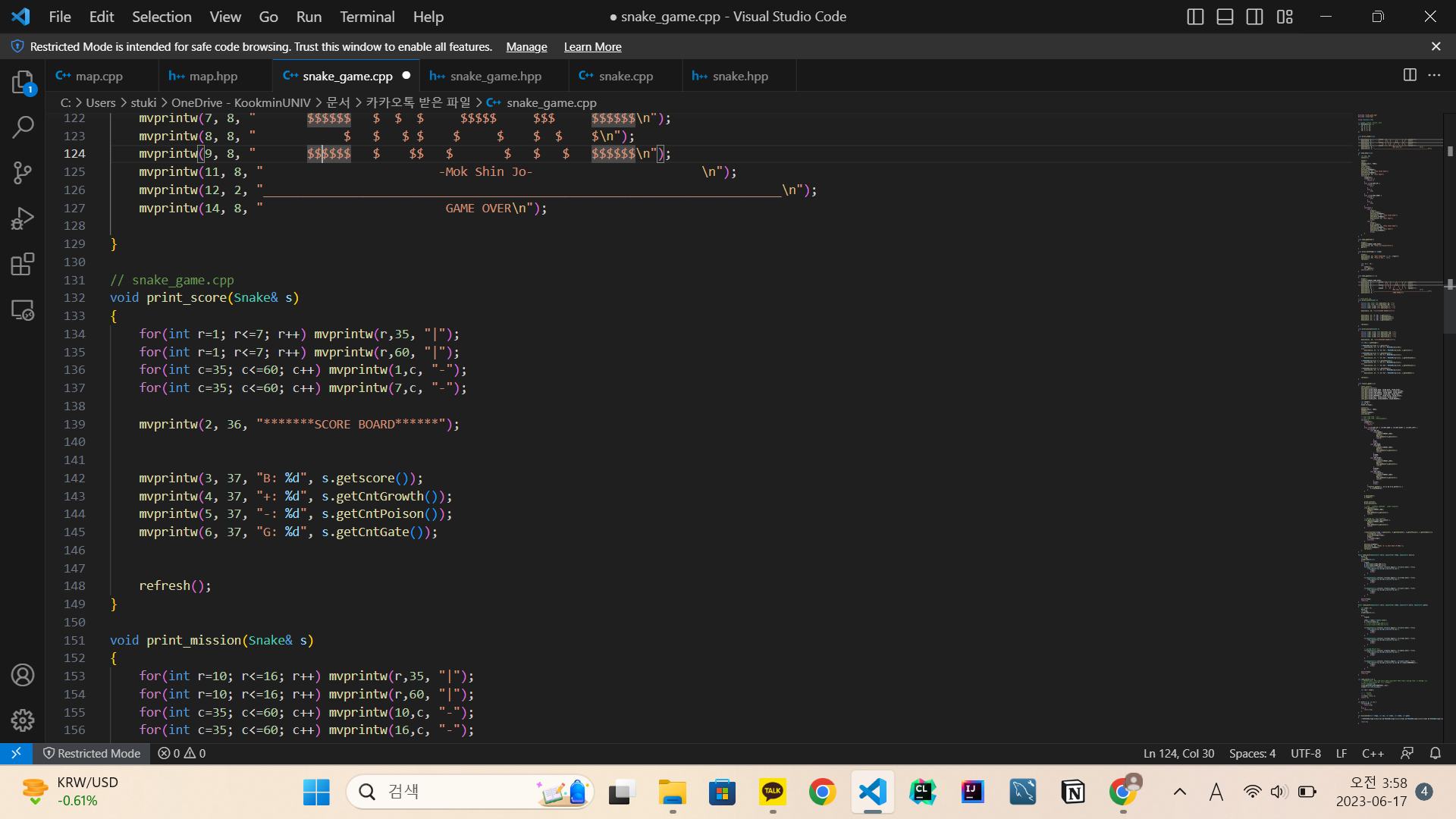
· 앞에서 설정한 기준값에 따라 Gate의 이동방향을 설정해준다.

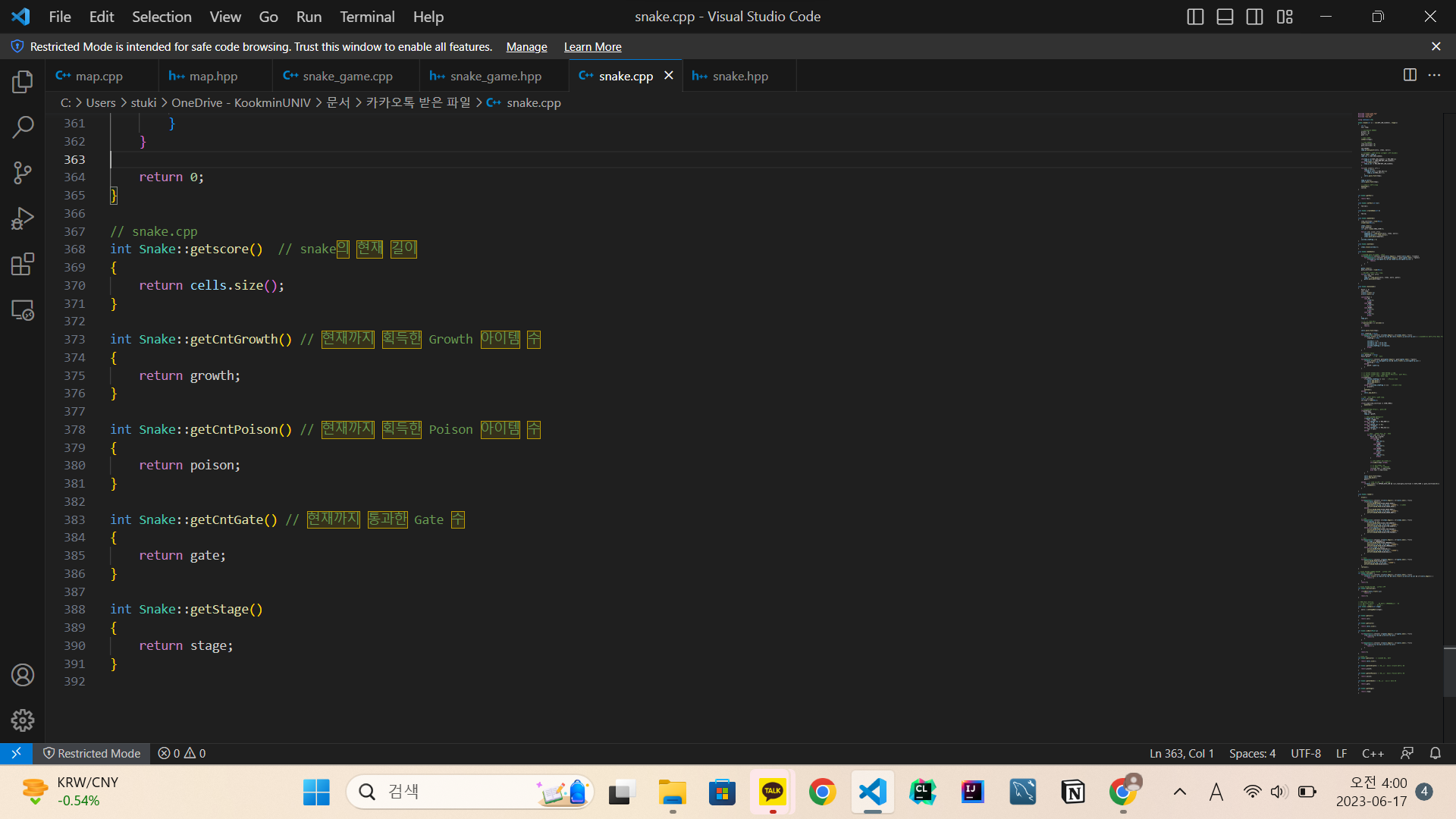
· gate통과 횟수를 저장하는 변수를 1 증가시킨다.

**5. ScoreBoard 표시 (개발자 : 신진욱)**

**5.1 print\_score()**

: 획득한 점수들을 화면에 표시해준다.





· Snake.getscore() : Snake의 현재 길이

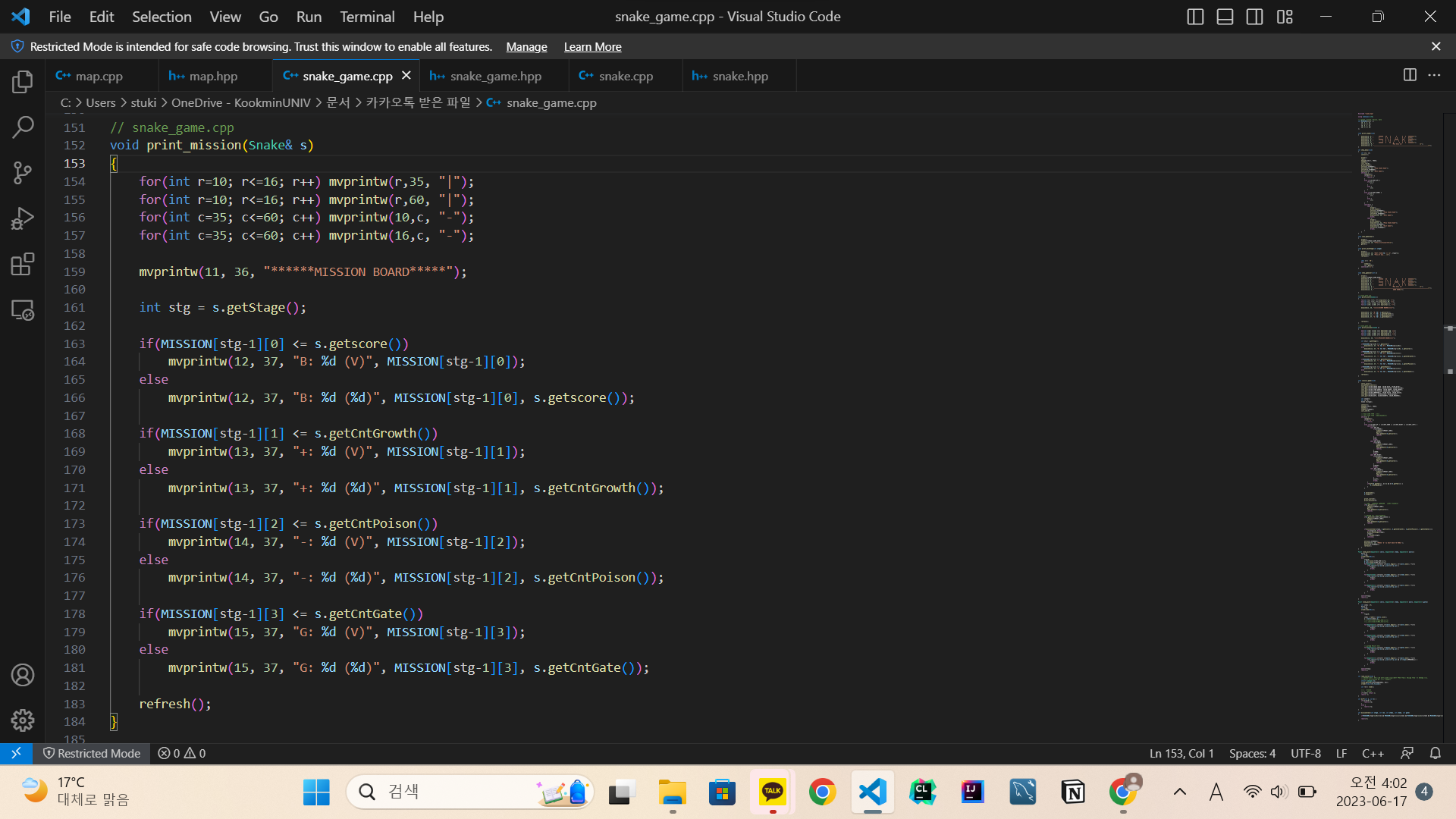
· Snake.getCntGrowth() : 현재까지 획득한 Growth 아이템 수

· Snake.getCntPoison(): 현재까지 획득한 Poison 아이템 수

· Snake.getCntGate() : 현재까지 통과한 Gate 수

**5.2 print\_mission()**

: 각 미션과 미션의 진행 상황 출력.

​​

### **2.2.3 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.** |
| --- |

**​​2.2.3.1 ncurses/ncursesw**

GUI기반 Game을 만들기위해 사용하였다.

Snake/Item/Gate를 Unicode 문자열로 표시할 때 사용한 라이브러리다.

### **2.2.3.2 deque(STL라이브러리)**

Snake/Item/Gate 좌표와 같은 정보를 담기 위해, 2차원 배열이 아닌 deque라는 STL라이브러리를 사용하였다. 특히, Snake 움직임을 구현하는 과정에서, push\_front같은 함수를 제공해주는 것 때문에 vector가 아닌 deque 컨테이너를 사용하게 되었다.

### **2.2.3.3 ctime**

random하게 좌표를 생성하기위해 ctime라이브러리를 활용하였다.

Item의 경우, 초단위로 생성하게 될 경우, 동시 출현 아이템들의 생성시간이 거의 같아서

종류가 모두 같아지는 버그가 발생하였다. 따라서 초단위가 아닌 millisecond 기준으로

seed를 생성하도록 clock\_gettime 함수를 사용하였다.

### **2.2.3.4 cstdlib**

rand() 함수를 호출하여 랜덤한 값을 추출하기위해 사용하였다.

### **2.2.4 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.** |
| --- |

- 각 Stage마다 Map을 구성할 때, Immunewall/wall에 대한 정보를 2차원 배열로 선언했었는데, 매번 2차원배열의 값을 수정하는 것 때문에 시간이 많이 소요되는 문제점이 있었다.

- 따라서, 2차원 배열이 아닌 deque에 해당 좌표를 저장하여 속도를 향상시킬 수 있도록 수정하였다.

- Snake의 Tail 방향으로 조작키를 클릭했을 때, 게임 오버 시키기 위해 이전 입력을 저장하였는데, 이때문에 다음 스테이지로 넘어가거나, Gate로 진입해 Snake의 진행 방향이 조작 버튼의 클릭과 관계없이 변화했을 때, 현재 진행 방향과 무관하게 마지막 로그에 따라 작동한다는 점이 제한적이었다.

- 마지막 조작을 저장하는 방식이 아닌, 현재 Snake의 진행방향을 저장하는 변수와 현재 입력되는 조작 버튼의 비교를 통해 동작하도록 수정함으로써 개선할 수 있었다.

### **2.2.5 결과물 목록**

**[ Youtube :** [**https://youtu.be/8c3oTD9Hqx4**](https://youtu.be/8c3oTD9Hqx4) **]**

**[ Github :** [**https://github.com/Jinhyup-Mok/C-\_Project**](https://github.com/Jinhyup-Mok/C-_Project) **]**

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

| **파일명** | **기능** |
| --- | --- |
| **Makefile** | 편리하게 컴파일할 수 있도록 함 |
| **main.cpp** | main 함수 호출 |
| **snake.hpp** | Snake Class의 선언 |
| **snake.cpp** | Snake Class 메소드 구현 |
| **snake\_game.hpp** | Game에 필요한 요소 선언 |
| **snake\_game.cpp** | Game에 필요한 핵심 로직 구현 |
| **map.hpp** | map.cpp 에서 사용되는 메소드 선언 |
| **map.cpp** | 각 stage마다 map을 구성하는 로직 구현 |

# 3. 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.** |
| --- |

| **이름** | **역할** | **느낀점** | **평가** |
| --- | --- | --- | --- |
| **목진협** | 프로젝트 UI 구성 / Snake 메소드 구현 / 시연 영상 제작 및 편집 | 프로젝트의 팀장을 맡으면서, 프로젝트를 완벽하게 완성할 수 있을지, 팀을 잘 이끌수 있을지에 대한 고민과 함께 첫 프로젝트 팀장이라는 기대감을 함께가지고 프로젝트를 수행했다. 프로젝트의 메인 UI를 팀원들과 함께 구성하고, 단점을 개선하는 과정에서 팀원간의 소통의 중요성과 오픈소스 라이브러리 활용에 대한 중요성을 느낄 수 있었다. 게임의 메인 로직을 구현하는 과정에서는 많은 난관과 시행착오를 겪었지만, 팀원들과 함께 고민하고, 코드를 계속해서 최적화하고 개선하는 과정에서 나 자신뿐만 아니라 팀원들과 함께 성장했음을 느낄 수 있었다. 완성한 프로젝트와 영상 편집을 마무리한 결과물을 팀원들과 함께 보았을때, 팀원 모두가 서로에게 고마움을 느끼고, 스스로에게는 뿌듯함과 자신감을 느낄 수 있는 좋은 기회가 되었음을 다시 한번 느끼게 되었다. | 9점 |
| **신진욱** | Score Board, Mission Board 구현 / Stage 및 Map 제작 / 보고서 작성 | 프로젝트를 수행하면서 수업 시간에 배운 헤더파일, 소스파일 분류를 경험해 볼 수 있었고, 추가로 메이크 파일도 만드는 기회도 접해봐서 좋았다. 프로젝트 UI를 구성하는 과정이 재밌었지만 맵, 보드를 원하는 위치에 배치시키기 위해서 자세한 값을 찾아가는 과정에서 조금 어려움을 겪었다. 또한 보드 값들이 다음 맵으로 넘어갈 때마다 초기화 해주는 것에서 어려움을 겪었다. 이러한 문제들이 있을 때 팀원들과 의사소통을 통해 해결했기 때문에 팀의 중요성을 느꼈다. | 9점 |
| **조민형** | 게임 Item 생성과 기능 구현 / Gate 생성 및 로직 구현 / 보고서 작성 | 아이템 생성과 Gate 로직에서 시행착오가 많아 어려웠다. 플레이어가 아이템을 먹어서 사라졌을 때와 시간이 지나 자동으로 사라졌을 때를 다르게 구현하는 것이 어려웠다. 예상치 못한 다양한 부분에서 난항을 겪었지만, 그럴 때마다 팀원들의 조언과 오픈소스를 참고하여 해결해나갈 수 있었다. 그러나 프로젝트를 진행함에 있어서 각자가 서로 다른 부분의 코드를 따로 작성하기 때문에 코드를 전체적으로 통일성 있고 깔끔하게 정리하는 데에 어려움이 있었다. 여러 사람과 협업하며 정해진 형식과 규칙에 맞게 역할을 수행해야할 중요성 또한 배웠다. | 9점 |

# 4. 참고 문헌

**참고한 서적, 기사, 기술 문서, 웹페이지를 나열한다.:**

| **번호** | **종류** | **제목** | **출처** | **발행년도** | **저자** | **기타** |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | 웹페이지 | NCURSES Programming HOWTO | https://wiki.kldp.org/wiki.php/NCURSES-Programming-HOWTO | 2014-3-22 | Pradeep Padala |  |
| **2** | 웹페이지 | cplusplus | https://cplusplus.com/ | 2000-2023 | x |  |
| **3** | 웹페이지 | Ncurses Programming Guide | https://jbwyatt.com/ncurses.html | x | x |  |
| **4** | 웹페이지 | c++ 이야기 | https://boycoding.tistory.com/category/C%2B%2B%20%EC%9D%B4%EC%95%BC%EA%B8%B0 | 2018-2020 | 소년코딩 |  |
| **5** | 웹페이지 | GNU ncurses | https://invisible-island.net/ncurses/announce.html | 2022-12-31 | Thomas E. Dickey |  |

# 

# 

# 

# 

# 

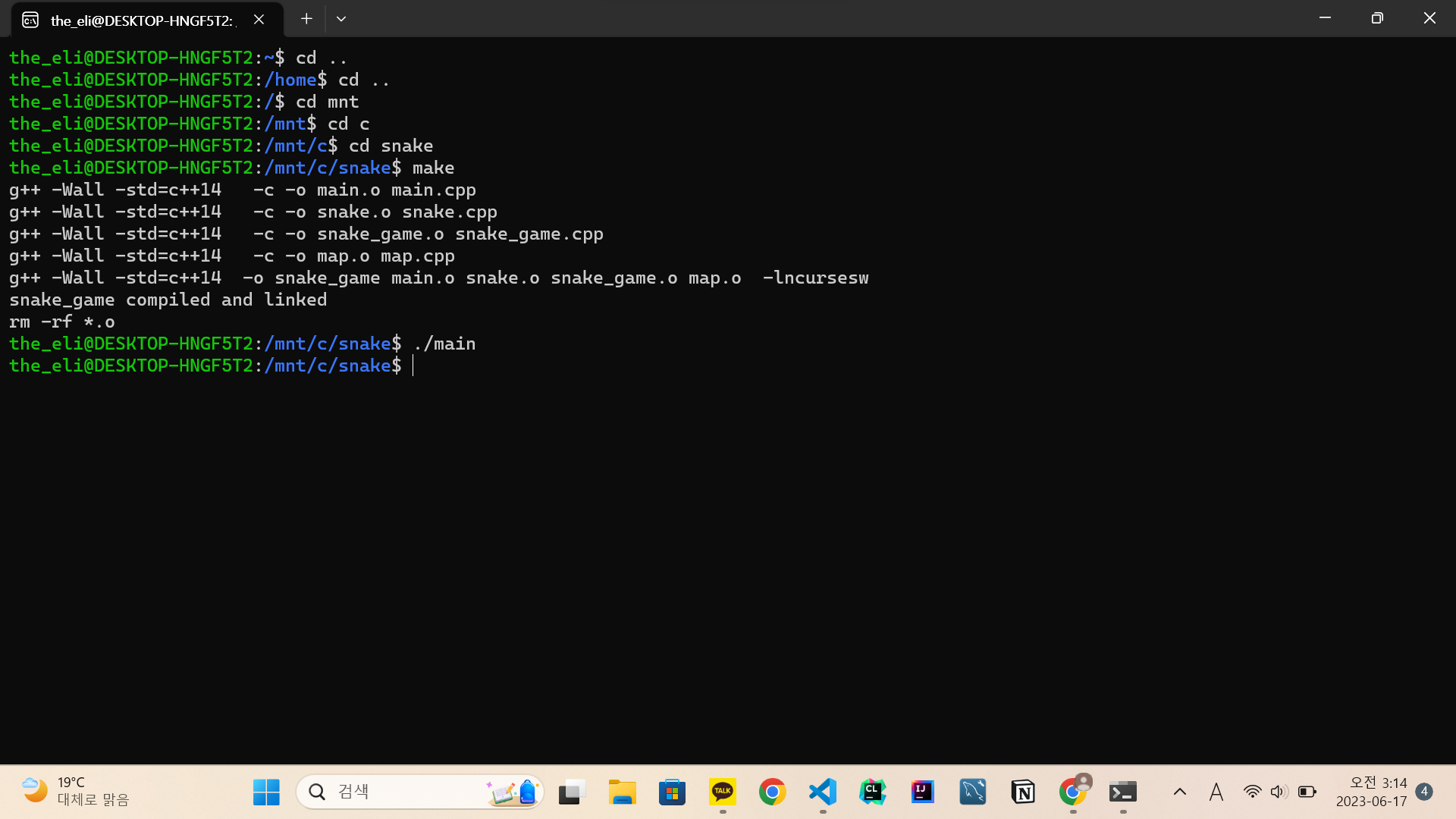
# 

# **5. 부록**

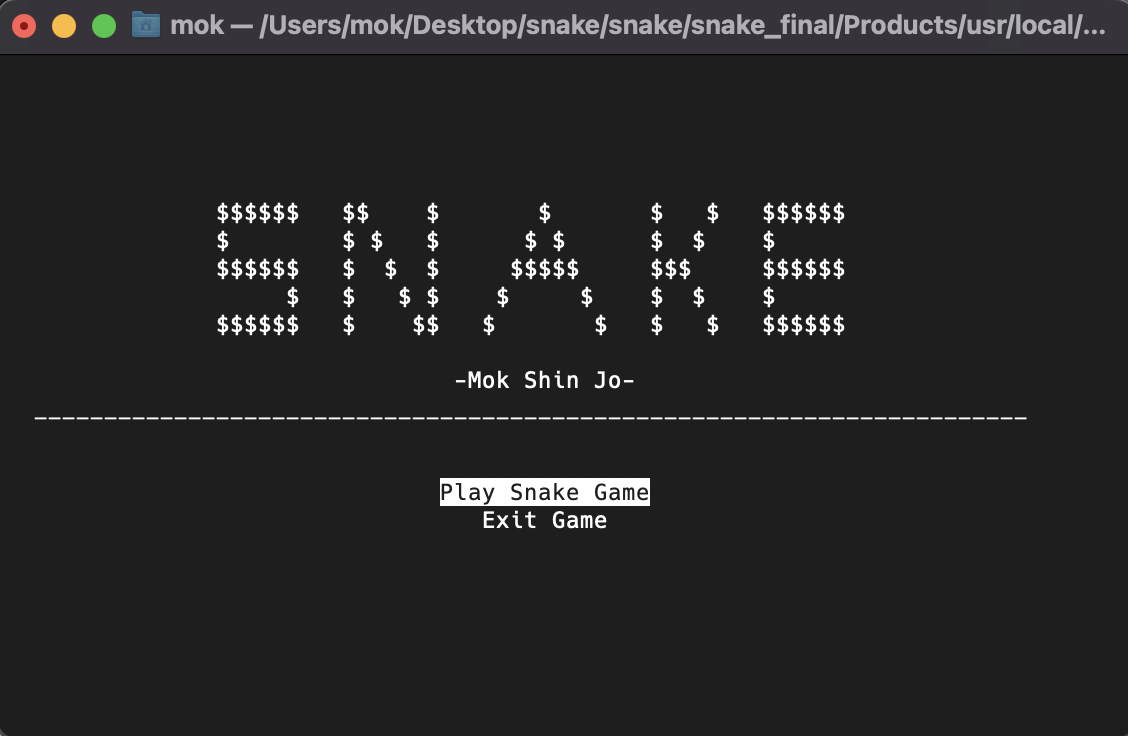
| **작성요령 (15점)**  **프로젝트의 결과물을 사용하기 위한 방법에 대해서 작성하세요.** |
| --- |

## **5.1 사용자 매뉴얼**

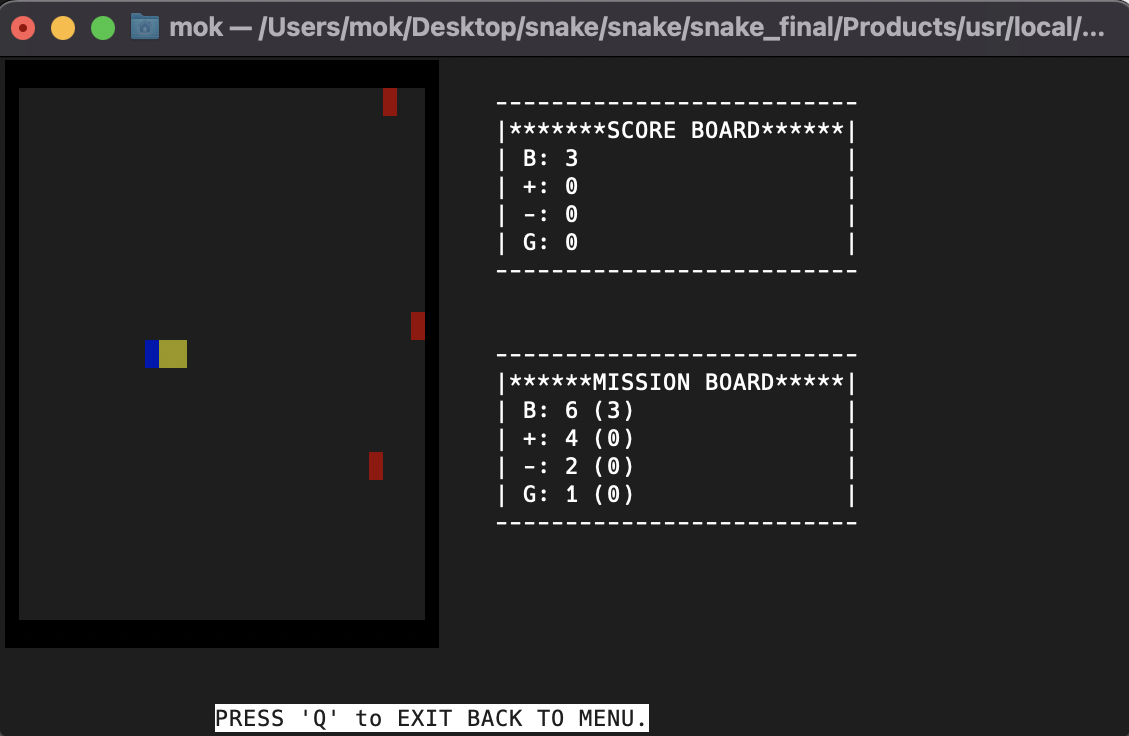
**프로젝트 실행 후 사용안내, 따라하기 등 포함**

****

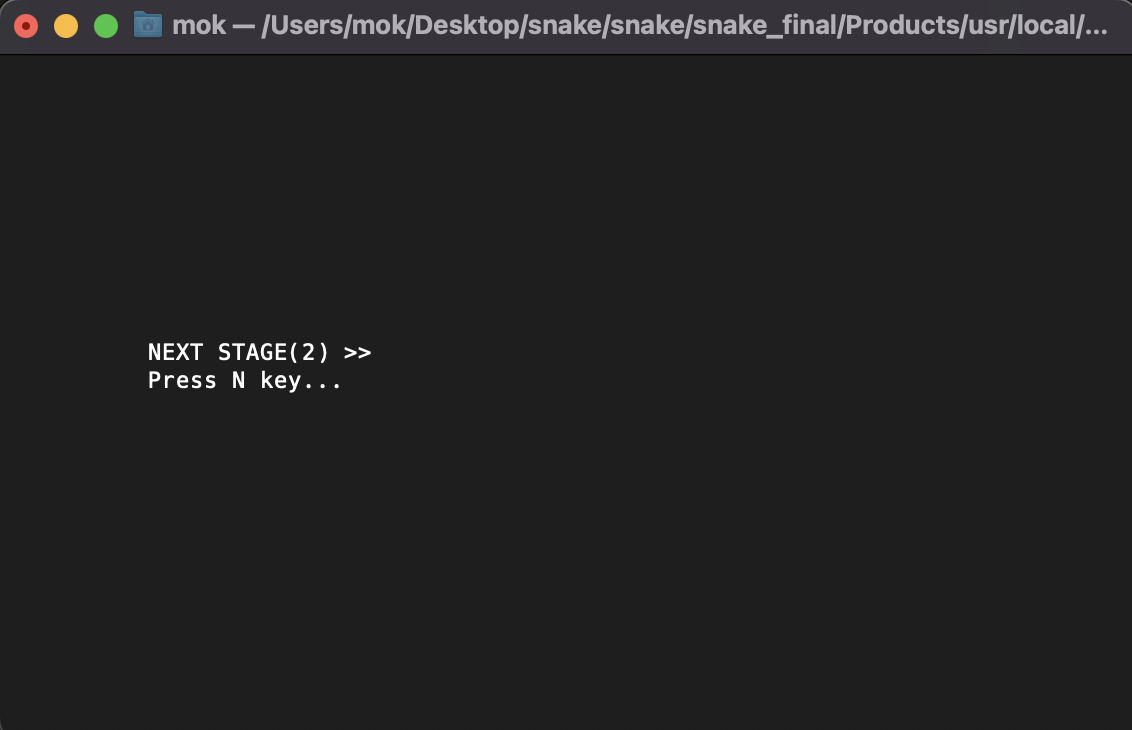
* 터미널을 연 후 cd 명령어를 이용하여 snake 폴더로 이동한다.
* make 명령어를 이용하여 소스 파일들을 copile, link 시켜준다.
* ./main을 입력하면 스네이크 게임이 실행된다.
* 실행 시 첫 화면. Play Snake Game을 엔터키로 누르면 게임이 시작된다.



* 게임 시작 첫 화면. 방향키를 이용하여 snake를 움직이며, 현재 이동 방향의 반대 방향으로는 못 간다.



* 초록색 item을 먹으면 몸의 길이가 1 증가하고, 보라색 item을 먹으면 몸의 길이가 1 감소한다.
* 하늘색은 gate이며 두개가 한 쌍으로 등장한다. 하나의 게이트로 들어가면 다른 하나의 게이트로 나오게 된다.
* 몸의 길이가 3보다 작아지거나 snake가 벽에 닿으면 게임이 종료된다.
* MISSION BOARD의 미션을 충족시키면 다음 stage로 넘어갈 수 있다.
* 미션을 통과하면 나오는 화면. ‘N’키를 누르면 다음 스테이지로 넘어간다.



* 게임을 종료하고 싶으면 ‘Q’키를 누르고 (첫 화면으로 돌아감) Exit Game을 엔터키로 누르면 된다.

## **5.2 설치 방법**

**프로그램 컴파일 및 실행 방법 포함**

* **5.1 참고**

### **5.2.1 컴파일**

* **개발OS :** Linux Ubuntu / Mac Os / Window
* **Version :** C++ 14
* **Compile Command :** 해당 Directory로 이동한 후**,** make

### **5.2.2 실행**

* **Command:** ./main