



국민대학교
소프트웨어융합대학
소프트웨어학부

C++프로그래밍 프로젝트

프로젝트 명	스네이크 게임
팀 명	스네이크
문서 제목	결과보고서

Version	1.3
Date	2022-06-15

팀원	이민석(팀장)
	오홍석
	석준환

CONFIDENTIALITY/SECURITY WARNING


 국민대학교 소프트웨어학부 C++프로그래밍	결과보고서		
	프로젝트 명		
	팀 명		
	스네이크	Version 1.3	2022-JUN-15

이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 C++프로그래밍 수강 학생 중 프로젝트 “스네이크 게임”을 수행하는 팀 “스네이크”의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 “스네이크”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역

Filename	최종보고서-스네이크게임.doc
원안작성자	이민석,오홍석,석준환
수정작업자	이민석,오홍석,석준환

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2022-5-31	이민석	1.0	최초 작성	개요, 목표, 개발내용 일부
2022-6-2	오홍석	1.1	내용 수정	개발 내용 및 결과물 일부, 나머지 작성
2022-06-05	석준환	1.2	내용 수정	활용 개발된 기술, 현실적 제한 요소, 해결 방안
2022-06-15	이민석	1.3	최종 수정	부록작성,

 국민대학교 소프트웨어학부 C++프로그래밍	결과보고서		
	프로젝트 명		
	팀 명		
	스네이크	Version 1.3	2022-JUN-15

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 9
 - 2.2.3 활용/개발된 기술 13
 - 2.2.4 현실적 제한 요소 및 그 해결 방안 14
 - 2.2.5 결과물 목록 16
- 3 자기평가 16
- 4 참고 문헌 17
- 5 부록 17
 - 5.1 사용자 매뉴얼 17
 - 5.2 설치 방법 18

 국민대학교 소프트웨어학부 C++프로그래밍	결과보고서		
	프로젝트 명		
	팀 명		
	스네이크	Version 1.3	2022-JUN-15

1 개요

평가기준 (10 점)

프로젝트를 완성하기 위해 사용한 개발 방법을 기술하세요.

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

C++언어를 사용하여 스네이크 게임 개발 과정에서 해결해야 하는 솔루션들은 크게 5 가지였다. map 구현, snake 움직임과 길이 구현, item 요소 구현, gate 구현, 점수판 구현 등이 있다. 첫 번째로 map 을 구현할 때는 배열을 사용했다. Int array 를 21*21 사이즈로 만들어서 안에 벽, 빈 공간(스네이크가 움직일 수 있는 공간), item, poison 등을 각자 숫자로 분리하여 구현했다.

그리고 나서 두 번째 스네이크를 구현했다. 스네이크를 구현할 때 동적 배열인 STL Vector 를 사용하여 구현했다. Snake 의 x 좌표와 y 좌표를 가지는 pair 들을 담는 벡터를 만들어 0 번째 index 를 헤드로 하는 스네이크를 만들고 매 틱마다 방향키에 맞게 움직인 좌표를 head 를 의미하는 0 번째 index 에 넣어주었다. 그리고 snake 의 좌표에 해당하는 map 의 숫자를 각각 head, body 를 의미하는 숫자로 변경해주면 snake 의 이동을 표현해 줄 수 있다. 만일 방향키가 반대가 되거나 head 좌표가 자신의 body 좌표와 같아지는 경우 혹은 헤드가 벽의 좌표와 겹치면 실행을 종료하게끔 만들었다.

세 번째로 아이템 구현에서는 item 들의 x, y 좌표를 가지는 pair 들을 담는 vector 를 만들어 item 들의 좌표를 관리하였고, 각각의 item 의 생성 시간을 관리하는 vector 를 만들어 item 이 일정 시간 지나도 존재한다면 사라지고 새로운 곳에 item 이 추가 될 수 있도록 만들었다. window 에 총 3 개의 item 이 등장하도록 만들었고, 해당 조건은 전역변수를 통해서 관리하였다.

네 번째로 gate 구현은 랜덤 함수를 이용해서 gate 의 x 좌표와 y 좌표를 구했다. 랜덤으로 생성된 x,y 좌표가 벽에 위치하는 경우에 gate 를 생성했다. Gate 에 들어갔을 때 출구 좌표를 2 개중 다른 하나의 게이트의 x,y 좌표로 저장해서 스네이크가 나오도록 했다. 또한 스네이크가 게이트 출구로 나왔을 때 방향을 설정해주어야 했다. 이때 벽의 위치에 따라서 방향을 정했고 만약 장애물(벽) 때문에 방향을 바꿔야 하는 경우에는 for 문을 이용해서 시계, 반시계, 반대방향 순차적으로 바꾸도록 설정했다.

다섯 번째로 점수판 구현에서는 난수 발생 함수를 통해서 목표를 설정하여줬다. 이것은 매 stage 마다 다르게 설정되며 mvprtw() 함수를 통해서 각각의 종목에 맞는 목표들을 window 에 표시해 주었고, 전역 변수를 통해서 item, gate, snake 길이, 시작 시간 등을 관리하여 window 에 표시해준다. 이를 통해 자신이 목표치까지 얼마나 남았는지를 알 수 있고, 각각의 목표 달성 여부를 확인 할 수 있다.

 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명		
	팀 명		
	스네이크	Version 1.3	2022-JUN-15


2 개발내용 및 결과물

2.1 목표

작성요령 (10 점)

프로젝트의 목표를 기술하세요. 각 단계별 목표를 구체적으로 쓰세요.

적용단계	내용	적용 여부
1 단계	map 을 구현하기 위해 우선 21*21 사이즈의 int array 를 준비하여 snake 와 item,poison 이 올라갈 공간을 만들어준다.	적용
2 단계	snake 는 매 시간(틱)마다 방향키에 따라 움직이는 기능 구현할 때 동적 배열로서 사용할 수 있는 STL 벡터를 사용하여 구현한다.	적용
3 단계	GrowItem, posionItem 합쳐서 3 개가 넘지 않게 한다. rand()함수를 통해서 item 의 좌표를 설정해준다. item 의 좌표가 snake 가 움직일 수 있는 공간이 아니라면 다시금 좌표를 생성한다. rand()함수를 통해서 growItem 을 추가할지, posionItem 을 추가하지 정해준다. 확률은 50:50 으로 한다.	적용
4 단계	gate 2 개 벽에 랜덤 생성하기 위해 rand 함수로 x 좌표 y 좌표를 2 개 구하고 그 좌표가 벽에 위치할 경우 그 벽을 gate 로 바꾼다	적용
5 단계	gate 통과시 출구 좌표를 먼저 저장하고 스네이크가 출구로 나왔을 때 방향을 설정한다. 나오는 위치, 장애물의 위치를 고려하여 나오는 방향을 결정한다.	적용

 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명		
	팀 명		
	스네이크	Version 1.3	2022-JUN-15

6 단계	<p>rand()함수를 통해서 목표치를 저장하는 전역 변수에 목표치를 할당해준다.</p> <p>mvprintw()함수를 통해서 각각의 목표치가 window 에 보일 수 있도록 설정한다.</p> <p>현재 달성치를 저장하는 전역변수를 통해서 목표치에 얼마나 가까운지 알 수 있게 표시한다.</p> <p>각각의 목표치 달성여부를 표시한다.</p>	적용
------	---	-----------

 <div> <p>국민대학교</p> <p>소프트웨어부</p> <p>C++프로그래밍</p> </div>	결과보고서		
	프로젝트 명		
	팀 명		
	스네이크	Version 1.3	2022-JUN-15

2.2 개발내용 및 결과물

2.2.1 개발내용

작성요령 (10 점)

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

1. 우선 맵을 구현하기 위해 21*21 사이즈의 int array 를 만들고 각각의 블록을 숫자로 역할을 나눈다.

0 : snake 가 움직이고 item 이 생성되는 공간

1: wall, gate 가 생성 가능하고 snake 의 Head 가 겹치면(닿으면) 게임이 종료된다

2: immune wall, gate 생성 불가능, snake 의 헤드가 겹치면(닿으면) 게임 종료된다

3: snake 의 헤드

4: snake 의 body

5: item

6: poison

7: gate

를 각각 표현하며 각각 다른 색으로 표현하기 위해 init_pair()를 사용하여 구현한다.

이렇게 각 스테이지를 만들고 클리어하면 기본 틀은 유지 한채 map 의 구성을 바꾸는 형태로 만든다.

또한 각 스테이지 사이에 리셋 기능을 넣어서 스테이지 명 표시, 맵에 있는 아이템,독,게이트 정리 등을 할 수 있는 함수를 만든다.

2. snake 는 매 시간마다 방향키에 따라 움직여야 하므로 동적 배열로서 사용할 수 있는 STL 벡터를 사용하여 구현한다. snake 의 x,y 좌표를 가지는 pair 들을 담은 vector 를 만들어 0 번째 인덱스를 Head 로 하는 snake 를 만든다.

매 틱마다 (아이템과 포이즌 배제) 방향키에 맞는(위로 = y-1,아래로 = y+1,좌 = x-1,우 = x+1)를 기존의 head 에다가 계산해 주어서 각각의 x,y 자리의 0 번째 인덱스에 넣어준다,

그 뒤 기존의 Head 는 첫번째 body 이 된다. 그리고 마지막 인덱스의 좌표를 지워주고 이렇게 만든 snake 의 좌표 해당하는 map 의 숫자를 각각

3(Head),4(body) 로 변경해주면 snake 의 이동을 표현할 수 있다.

 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명		
	팀 명		
	스네이크	Version 1.3	2022-JUN-15

또한 방향키가 반대가 되거나 Head 의 좌표가 자신의 body 의 좌표와 같아지거나 Head 가 벽의 좌표와 겹치게 되면 실행을 종료하게 만든다.

3. posion item 과 grow item 의 개수를 합쳐 총 3 개가 넘지 않도록 하기 위해서 전역변수를 생성해 총 item 의 수를 저장한다. item 들의 x,y 좌표를 저장하는 pair 들을 담은 vector 를 생성하여 posion 과 grow item 을 관리한다. 각각의 item 들의 제한시간을 저장하는 vector 를 만들어서 관리해준다. 만약 item 의 제한시간이 0 이 된다면 해당 item 의 x,y 좌표를 제거해주고, 새로운 item 의 좌표를 할당해 추가해준다.

grow_item 을 snake 의 Head 가 마주쳤을 때 snake 의 좌표를 담은 vector 의 뒤에 좌표를 추가하여준다. 그래서 전체 길이를 늘려주는 방식이다.

posion_item 을 snake 의 Head 가 마주쳤을 때는 map 에서 snake 의 꼬리 부분의 좌표값을 빈 공간으로 설정해주고 snake vector 에서 꼬리 부분 좌표를 pop_back()해준다.

item 을 먹으면 총 아이템 개수를 줄이고, item 을 만드는 함수를 호출한다.

4. timer 를 이용해서 게임시간이 10 초가 지났을 Gate 2 개를 벽에 랜덤으로 생성한다. rand 함수로 x 좌표, y 좌표를 2 개 구하고 그 좌표가 벽에 위치할 경우 그 벽을 gate 로 바꾼다. 그리고 나서 gate 를 통과했을 때 움직임을 구현한다. 예를 들어 gate1 으로 통과하는 경우 출구 좌표는 gate2 의 x,y 좌표로 설정해준다. 스네이크가 출구로 나왔을 때 방향의 우선순위는 다음과 같다. 1 순위는 진입 방향 그대로 나가는 것이고 2 순위는 시계 방향 회전 3 순위는 반시계 방향 4 순위는 진행 방향의 반대로 방향을 설정하는 것이다. 따라서 위의 4 경우를 for 문을 통해 순차적으로 방향을 설정하고 그 방향에 장애물이 없는 경우 break 문을 통해 for loop 를 벗어나서 스네이크가 그 방향으로 이동하도록 한다.

5. rand()함수를 통해서 snake 의 최대 길이, 각각의 item 획득 횟수, gate 통과 횟수 등의 목표치를 할당해준다. mvprintw()함수를 이용하여 window 에 각각의 목표치들을 확인할 수 있게 표시해주고, 현재 진행 상황을 저장하는 전역 변수들을 만들어 window 에 표시되어지는 달성치를 계속 갱신해준다. 만약 목표치를 달성하면 check 표시를 통해서 해당 목표치를 달성했음을 표시해주고, 모든 목표치를 달성 완료하면 다음 스테이지로 넘어간다. 다음 스테이지로 넘어가면 다시 rand()함수를 통해서 새로운 목표치를 설정하여 준다.

 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명		
	팀 명		
	스네이크	Version 1.3	2022-JUN-15

2.2.2 시스템구조 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.

1,2 이민석

3,5 오홍석

4 석준환

1.Map , 스테이지 초기화 설정.

스테이지별 맵 설정, 윈도우 설정

게임을 시작하면 void reset() 함수를 실행한다.

initscr()로 NCURSES TUI 모드 시작

start_color() 로 색깔모드 사용 선언

init_pair(a,b,c) 로 팔레트에 b 폰트색,c 배경색 적용, 이걸 폰트와 배경을 같게 해서 블럭처럼 보이게 한다.

나중에 attron(COLOR_PAIR(a)) 형태로 사

noecho() 로 입력받은키 출력 막기

curs_set(0) 로 마우스 커서 숨기기

cbreak() 입력받은키 바로바로 적용

keypad(stdscr,TRUE) 로써 키 입력을 받아온다

getmaxyx(stdscr,winy,winx)로 winy,winx 에 표준화면 stdscr 의 가로 세로 값을 넣어준다.

이렇게 게임을 위한 기본 설정후 setstage() 함수로 넘어가서 게임 실행을 위한

movetimer,edible_timer,gatecounter 를 초기화하고(매판 초기화 되는 전역변수이다)

map[a][b][c] 는 a 단계만큼의 스테이지를 B*C 형태로 표현해준다

여기서 enum 을 사용하여 immunewall,wall,snake_head,snake_tail,

item,poison,field 를 각각 구현해준다.

map 에 있는 번호를 탐색하여 맵상의 객체들을 attron(COLOR_PAIR()) 이용해

번호에 해당하는 색을 입혀준다.

 국민대학교 소프트웨어학부 C++프로그래밍	결과보고서		
	프로젝트 명		
	팀 명		
	스네이크	Version 1.3	2022-JUN-15

처음 뱀의 위치를 각각 스테이지의 맵에 맞게 만들어준다, 스테이지 별로 벽의 위치가 다르므로

처음 움직임에 지장이 없도록 지정하여준다.

Void check_target() 에서

만약 스테이지 1,2,3,일때 스테이지를 클리어하면 다음단계로 넘어가야 하므로

clear()를 이용해 윈도우를 모두 지우고

mvprintw(a,b,문자열),refresh() 이용하여 (b,a) 위치에 문자열을 띄워주며

다음 스테이지 setstage(stage++) 로 이동해야한다.

스테이지 4 일때는 게임이 끝나므로 중지시켜야한다 따라서 전역변수 fail 을 true 로 바꾸고 all_clear 도 true 로 바꾸어 "you fail"문구를 띄우지 않고 바로 윈도우를 종료시킨다.

2.snake 의 구현 및 일반적인 상황에서의 snake 움직임(방향고정 or 방향키를 이용한 조작, gate 이용하여 방향이 바뀌는건 여기서 다루지 않음) (game rule #1)

snake 는 벡터로 구현한다. 매 틱마다 움직여야하므로(move)함수 구현) 매 틱마다 배열에 관한 연산이 일어나야 한다 따라서 계산이 상대적으로 간편한 벡터로 구현하였으며 x,y 값을 모두 관리해야 하므로

vector<pair<int,int>>로 만들고 first 값에 y, second 값에 x 를 매칭시켜준다.

매 틱마다 전진을 하게 되는데

만약 스네이크의 길이가 3 이라고 하고 편의상[x,y] 라고 표현하자면(각각 x 좌표의 길이 3 벡터, y 좌표의 길이 3 벡터)

스네이크는 움직인다고 하더라도 Head 와 tail 만 변화가 생기게 되고 (아이템이 없는 상황이라면 테일은 줄어들기만 한다) 따라서 Head 는 유지하고 나머지 뒤에 것들은 각각 앞의 것을 복사 해 주면 된다.

```
for(int l = last - 1; l > 0; l --){x[l]= x[l-1]; y[l]= y[l-1]}
```

(last 는 현재 snake 의 길이이다, snake.size()) 처럼 해주면 Head 는 그대로 유지되지만 body 는 각각 하나씩 뒤로 밀린 형태가 된다.

또한 현재 Head 에 상태는 움직이기 전의 상태의 Head 와 같다 따라서 헤드에 방향에 해당하는 값을 연산하여주어야 한다. 스네이크는 방향키를 조작해 움직이더라도 단 한 칸씩 밖에 움직일 수가 없다. 따라서 만약 방향이 Up 인 상태에서 move 가 호출되면 snake[0]의 first 값에 -1 을 해주어야 한다. 마찬가지로 Down 일때 first 값에 +1, Left 일때는 second 값에 - 1, Right 일때 second 값에 + 1 을 해주어야 한다,

그 후에 map 에다가 수정된 좌표를 전달하여 이동을 표현한다.따라서 vector<pair<int,int>>direction 에 각 방향 별로 x,y 에 연산해줘야 하는 값을 저장한다.

 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명		
	팀 명		
	스네이크	Version 1.3	2022-JUN-15

방향키 입력에 따라 direction 에 저장되어있는 {0,1},{1,0},{-1,0},{0,-1}을 선택하여 매 move()마다 Head 에 더해준다.

방향을 나타내는 enum 을 만들고 up,down,left,right 를 넣는다. 해당 enum 은 direction 의 index 로 활용할 수 있게 순서를 맞춰서 넣어준다.

현재 방향기와 입력 방향키가 거꾸로 가는 경우에 'fail'이 되게 하고 나머지는 dir 이 enum 값에 해당하는 up,down,left,right 을 가지게 한다. 따라서 snake[0].first += direction[dir].first;

snake[0].second += direction[dir].second;같은 방향에 따른 머리의 값 변화를 나타내었다.

방향은 input 에서 getch() 로 처리했다. getch() 값이 KEY_UP 이면 dir 은 UP 을 연결 시킨다. 만약 반대되는 입력 (ex KEY_UP 인데 현재 방향이 down) 이라면 fail 을 true 로 바꾸고 다음 턴에 게임을 끝낸다.

3.

item 생성 알고리즘 - 전역 변수 item_Cnt 를 통해서 전체 item 개수를 조절하여준다. 만약 item_Cnt 가 3 보다 작다면 while 문에 진입하게 된다. Rand()%19 + 1 을 통해서 map 좌표에서 벽을 제외한 공간의 좌표를 얻어내고 만약 해당 좌표 값이 FIELD 로 되어 있지 않다면 item 을 생성할 수 없는 위치이기 때문에 continue 를 통해서 다음 반복을 진행하고 FIELD 라면 해당 좌표 값을 item 좌표로 설정하여 item vector 에 추가해준다. 후에 item_timer vector 에 item 의 제한 시간을 추가하여 준다. item 의 종류는 rand()함수를 이용하여 랜덤으로 grow, posion 이 들어가게 하였다. 확률은 50 대 50 으로 한다.

Item 제한 시간 초과 시 - timer()라는 함수를 통해서 item 의 제한 시간을 관리한다. timer()함수는 매 턴마다 호출되어지는데 호출되어질 때마다 item_timer 에 저장되어 있는 제한 시간을 1 씩 빼준다. 그 후 만약 item_timer 에 있는 제한 시간 중에 0 인 것을 발견한다면 해당 timer 의 item 은 item vector 에 저장된 좌표를 이용하여 map 에 FIELD 로 바꾸어주고 item_timer 와 item vector 에서 erase 함수를 통해 제거해준다. 또한 전역 변수인 item_Cnt 의 개수도 1 줄여준다. item 제거 후 다시 item 생성을 위해 makeItem()함수를 호출하여준다.

Item 획득 시 - 만약 snake 가 이동 중에 snake 의 Head 부분이 map 상에 표현되어있는 ITEM 과 겹친다면 item 을 획득한 것으로 간주한다. 따라서 snake 의 Head 와 body 좌표를 저장하고 있는 vector 의 뒤에 push_back()함수를 이용하여 body 를 추가하여준다. 그 후 grow item 을 획득한 것이기에 chk_grow 변수에 + 1 을 해준다.

또는 ITEM 이 아니라 POSION 과 겹친다면 posion item 을 획득한 것으로 간주한다. Grow item 과 다르게 길이가 줄어드는 item 이므로 snake 의 길이가 3 이라면 fail = true 를 통해 다음 턴에 게임을 종료시킨다. 만약 길이가 3 보다 크다면 map 에 표시되어있는 snake 의 꼬리 좌표 부분을 FIELD 로 바꾸어준다. 그 후에 snake 의 Head 와 body 좌표가 저장되어 있는 vector 에서 pop_back()함수를 이용하여 꼬리 부분의 좌표를 제거하여준다. Posion item 도 동일하게 현재 달성치를 확인해주는 변수인 chk_posion 변수에 + 1 을 해준다.

만약 snake 가 움직이다가 item 을 먹은 경우 새로운 item 을 추가 해주어야 하기 때문에 getItem 변수를 선언하여 item 을 먹으면 1 로 초기화해준다. 만약 getItem 변수가 1 이라면 grow item 이나 posion item 을 획득 한 것이기 때문에 item vector 을 돌면서 해당 좌표 값이 SNAKE_HEAD 인 것을 vector 에서 제거하여준다. item_timer 도 동일하게 제거하여준다. Item 개수가 줄어들었으므로 item_Cnt 변수에 -1 을 해주고, makeItem() 함수를 호출하여 새로운 아이템을 추가하여 준다.

 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명		
	팀 명		
	스네이크	Version 1.3	2022-JUN-15

4.-두개의 게이트 생성 알고리즘

for 문을 이용하여 2 번 반복하여 gate_x, gate_y 좌표를 생성하고 그 좌표가 벽에 위치할 때 pair 로 묶어서 door 벡터에 저장한다. 이 알고리즘은 Timer 를 사용해서 game 시간이 10 초 지났을 때 실행되도록 설정한다.

-게이트 통과시 알고리즘

스네이크가 게이트로 들어갔을 때 출구는 또 다른 하나의 게이트로 설정한다. 먼저 door 라는 vector pair 에 게이트 좌표값을 저장했다.

door 의 0 번째 index 를 출구로 설정했을 때 스네이크의 Head 부분과 겹치지 않는다면 그대로 출구로 지정했고 만약 겹친다면 1 번째 index 값의 x,y 좌표를 출구로 지정했다. 그리고 게이트 진출 방향을 설정한다.


int array[4] = {0,1,3,2} 이 코드는 출구로 스네이크가 나왔을 때 방향의 우선 순위이다. 0 은 뱀의 진출방향과 같은 방향이고 1 은 시계방향을 가르키고 3 은 반 시계방향, 2 는 반대 방향을 가르킨다. (dir + array[i]) % 4 를 통해서 진출 방향을 정해준다. 정해진 방향으로 진출 좌표를 설정하고 진출 좌표가 map 의 범위를 벗어나지 않고 WALL, IMMUNE_WALL, GATE 가 아니라면 해당 좌표를 스네이크의 Head 좌표로 설정하고 dir 을 (dir + array[i])%4 로 설정하여 준다. 후 break 문을 통해서 반복문을 탈출한다.

따라서 이를 for 문으로 실행해서 우선순위를 바탕으로 방향을 매 순간마다 바꾼다. 방향을 바꿨을 때 아무 장애물이 없다면 그때의 방향을 최종 방향으로 결정하는 것이다.

5. score board 생성

setstage()함수가 실행 될 때 setgoal()함수를 통해서 목표치가 설정된다. 목표치를 저장하기 위해 target_len, target_grow, target_posion, target_gate 라는 전역 변수들을 선언한다. 그리고 setgoal()함수에서 rand()함수를 통해 각각의 전역 변수들에 목표치를 할당하여 준다. 이는 매 stage 마다 새롭게 갱신되어진다. 그리고 현재 달성치를 저장하는 len_body, chk_grow, chk_posion, chk_gate 를 선언하여 준다. 해당 변수들은 각각의 item 을 먹거나 gate 를 통과하면 1 씩 늘어나게 설정되어 있다.

점수판은 show_window()함수에서 표시되어지는데 mvprintw() 함수를 이용하여 설정한 좌표 값에 lable 과 함께 할당된 목표치를 표시해준다. 그 밑에는 게임 시작 시간을 초 단위로 표시하여 준다. 또한 현재 달성치를 보여주기 위해서 score board 밑에 Mission board 를 생성하여 현재 달성치를 저장하는 전역 변수들을 표시하여 준다. 만약 달성치가 목표치와 같거나 크다면 해당 목표치를 달성한 것이기 때문에 check 표시를 통해서 목표 달성 여부를 표시하여준다. 만약 모든 목표치를 달성한다면 해당 stage 를 클리한 것이므로 clear() 함수를 통해서 window 를 정리하여 주고 stage 통과 label 를 mvprintw()를 통해

 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명		
	팀 명		
	스네이크	Version 1.3	2022-JUN-15

표시하여준다. 후에 stage 가 3 보다 작다면 stage++을 해주어 setstage(stage)를 해준다. stage 가 3 보다 작지 않다면 모든 stage 를 clear 한 것이기 때문에 전역변수인 all_clear 를 true 로 설정해주면서 다음 틱에 게임을 종료한다.

2.2.3 활용개발된기술

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.

gate 개발 과정에서 사용된 함수는 rand 함수, array 와 for 문등을 이용했다
rand 함수로 게이트의 위치를 설정했고, array 를 이용하여 우선 순위대로 출구로 나왔을 때의 방향을 설정한다.

enumeration 을 통해서 가시성을 극대화 시킨다.

Enum {FIELD, WALL}; 이라면 FIELD 는 0, WALL 은 1 이다.

 국민대학교 소프트웨어학부 C++프로그래밍	결과보고서		
	프로젝트 명		
	팀 명		
	스네이크	Version 1.3	2022-JUN-15

방향 별로 snake 의 Head 에 더해주어야하는 x,y 값을 pair 로 저장하여 방향에 맞게 할당해주어 code 를 간략화하였다.

gate 진출 방향 결정에서도 진입방향일 때, 시계방향 일때, 반시계방향일때, 반대방향일때 각각 더해주어야 하는 index 순서로 배열을 만들어 하나의 반복문을 통해서 4 가지의 상황을 처리하여주었다. (dir 은 현재 진행 방향이다.) 진입방향 일 때 $(dir+0)\%4$ 이다. 시계방향일때 $(dir+1)\%4$ 이고, 반시계방향일때는 $(dir+3)\%4$, 반대 방향일때 $(dir+2)\%4$ 이다. 이러한 방식으로 진출 방향을 설정하고 해당 좌표가 map 의 index 에 적합하고 WALL, IMMUNE_WALL, GATE 가 아니라면 해당 좌표를 snake 의 Head 의 좌표로 설정하고 dir 을 $(dir+array[i])\%4$ 로 설정하여준다. 후 break 를 통해 반복문을 탈출

Snake 구현을 위한 `vector<pair<int,int>>` - snake 의 x,y 좌표를 같이 관리하기 위해서 pair 를 저장하는 vector 를 이용

`initscr()`로 NCURSES TUI 모드 시작

`start_color()` 로 색깔모드 사용 선언

`init_pair(a,b,c)` 로 팔레트에 b 폰트색,c 배경색 적용, 이걸 폰트와 배경을 같게 해서 블럭처럼 보이게 한다.

`attron(COLOR_PAIR(a))` 로 색깔 입히기.

`noecho()` 로 입력받은키 출력 막기

`curs_set(0)` 로 마우스 커서 숨기기

`cbreak()` 입력받은키 바로바로 적용

`keypad(stdscr,TRUE)` 로써 키 입력을 받아오기

`getmaxyx(stdscr,winy,winx)`로 winy,winx 에 표준화면 stdscr 의 가로 세로 값을 넣어주기(나중에 문자 출력)


`printw()` - curses 모드가 시작되면 `printf()`와 같은 일반 출력함수로는 화면에 출력할 수 없어서 `printw()`를 사용한다.

`mvprintw(a,b,문자열),refresh()` 으로 윈도우에 문자 출력.

2.2.4 현실적제한 요소및그해결 방안

작성요령 (5 점)

제안된 프로젝트의 단계 별 수행에 있어, 제한 요소를 찾아 작성한다. 해당 제한 요소를

 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명		
	팀 명		
	스네이크	Version 1.3	2022-JUN-15

해결하기 위해서 어떤 방법으로 해결하였는지 작성한다.

게이트에서 나왔을 때 장애물이 있는 경우 방향 설정을 if 문으로 모든 경우의 수를 사용해서 작성하였더니 코드의 길이가 너무 길었다. 그래서 이 문제를 해결하기 위해 출구로 나오고 나서의 방향의 우선순위를 결정했고 for 문으로 우선 순위대로 방향을 설정한다. 만약 그 과정에서 맞는 방향을 찾았다면 그때 loop 를 나와서 그 방향으로 나오도록 했다. 또한 종강후 하루만에 제출해야해서 시간적으로 너무 촉박했다. 또한 STL 배우기전에 이 고제를 받게 되어서 처음에 어떻게 구현해야 할지 감이 잡히지 않았다.

 국민대학교 소프트웨어학부 C++프로그래밍	결과보고서		
	프로젝트 명		
	팀 명		
	스네이크	Version 1.3	2022-JUN-15

2.2.5 결과물 목록

Main.cpp

규모가 크지 않은 코드여서 헤더와 클래스 파일을 구분 하지 않았습니다.

3 자기평가

석준환-프로젝트를 하면서 어려움을 느끼는 순간들이 종종 있었다. 일단 첫째로 코드를 작성하고 다른 팀원이 만든 코드에 붙여서 오류가 발생하는 것이 힘들었다. 그럴 때마다 팀원들과 합심하여 코드 오류를 수정했다. 두번째로 혼자서 게이트 부분 코드를 설계하고 작성하는 것이 어려웠다. 특히 게이트를 나오고 나서 snake의 움직임을 어떻게 구현할 지에 대해 어려움을 느꼈다. 시행착오를 겪으며 그 부분을 무사히 구현했다. 프로젝트를 팀 단위로 하면서 느낀 어려운 점들과 코드를 설계 구현하는 점에서 어려움을 느꼈지만 결국 잘 마무리해냈다. 그러한 점에서 좋은 점을 둔다. 그러나 스스로 아쉬웠던 점은 오류를 찾아내는 것이 너무 느렸다. 디버깅을 빠르게 해야 다음 스텝으로 넘어갈 수 있는데 그러지 못해서 아쉬웠다.

이민석 - 처음에 STL 을 배우기 전에 이 과제를 받았을 때는 너무나도 막막했다. 처음에 어떻게 접근해야 할지도 모르겠고 답답하기만 했는데 수업시간에 STL 을 배우면서 점점 어떻게 해야 할 감이 잡히면서 점점 과제가 재미있어졌다. 물론 어렵고 모르는 부분도 많았다. 처음에 실행할 때는 실행도 되지 않고 컴파일도 되지 않을 때가 많았다 그러다가 점점 맵 이 나오고, 스네이크 가 움직이고, 블럭이 먹어지고 할 때마다 굉장히 보람차고 재미있었다. 팀 과제가 확실히 규모도 크고 어렵긴 하지만 그만큼 결과물을 만들어냈을 때의 성취감은 훨씬 커지는 것 같다.

오흥석 - item 요소의 구현, 점수 요소의 구현을 맡아서 진행하였다. 개발 초기에 각자 맡은 부분에 대해 코드를 작성해와서 연결하였는데 변수명, 함수명, 좌표 설정 부분 등 많은 부분에서 오류가 발생하였는데, 각자 작성한 코드의 스타일이 다르고 생각이 다르다 보니 어려움이 많았다. 그렇지만 팀원들과 머리를 맞대 고민하다보니 하나씩 해결해 나갈 수 있었다. 또한 Gate 요소의 코드 부분을 함께 수정해가면서 코딩 능력을 향상 시킬 수 있었고, 다른 사람의 코드를 분석하는 것이 쉽지 않다는 것을 느낄 수 있었다. 프로젝트를 진행하면서 다른 사람과 진행하는 프로젝트가 얼마나 어려운 것인지 알게 되었고, 자신의 의견을 관철해나가는 것만이 능사가 아니라 다른 사람의 의견 또한 존중하고 맞춰나가는 것이 중요하다는 것을 알게 되었다. 수업시간에 배웠던 STL 부분이 많은 도움이 되었다. 아쉬웠던 점은 객체 지향 프로그래밍을 하고 싶었지만 처음 구조를 설계할 때 그렇게 하지 못해 코드를 수정하는 과정에서 큰 어려움을 겪었다. 다음에는 객체 지향적 프로그래밍을 통해서 더욱 완성도 있는 프로젝트를 진행하고 싶다.

 국민대학교 소프트웨어학부 C++프로그래밍	결과보고서		
	프로젝트 명		
	팀 명		
	스네이크	Version 1.3	2022-JUN-15

4 참고 문헌

참고한 서적, 기사, 기술 문서, 웹페이지를 나열한다.:

번호	종류	제목	출처	발행년도	저자	기타
01	웹사이트	NCURSES wiki				

5 부록

작성요령 (15 점)

프로젝트의 결과물을 사용하기 위한 방법에 대해서 작성하세요.

5.1 사용자 매뉴얼

방향키를 사용하여 snake 의 헤드 방향을 결정합니다.

스네이크는 매 틱마다 전진을하고

게이트는 10 초후 등장합니다.

초록 블록을 먹으면 길이가 1 개 길어지고,

빨간 블록을 먹으면 길이가 1 개 작아집니다.

만약 길이가 3 아래로 떨어지면 게임에서 패배하게됩니다.

또한 snake 의 헤드가 바디에 닿거나, 헤드가 게이트가 아닌 벽에 닿거나

혹은 이동방향과 반대되는 방향키를 누르면 패배하게 됩니다.

점수는

1. 몸의 최대 길이.

2. 게임중 획득한 item 개수.

3. 게임중 획득한 poison 개수.


4. 게임중 gate 사용횟수.

5. 게임 시간.

입니다.

옆의 점수표의 미션을 모두 달성하면 다음 스테이지로 가게 됩니다.

스테이지는 총 4 개 입니다.

 국민대학교 소프트웨어부 C++프로그래밍	결과보고서		
	프로젝트 명		
	팀 명		
	스네이크	Version 1.3	2022-JUN-15

5.2 설치방법

NCURSES 를 설치해야하므로

```
$ sudo apt-get update
```

```
$ sudo apt-get install libncurses5-dev libncursesw5-dev
```

를 하여 NCURSES 를 설치하여 줍니다.

실행은 파일이 하나임으로 g++ main main.cpp -lncursesw 로 컴파일 하여 실행한다.

```
# Makefile
```

```
CC = g++
```

```
CCFLAGS = -g
```

```
all : main.o
```

```
$(CC) $(CCFLAGS) -o main main.o -lncursesw
```

```
clean:
```

```
rm -f *.o
```

```
%o : %.cpp
```

```
$(CC) $(STD) $(CCFLAGS) -c $<
```

해당 Makefile 을 이용하여 실행파일을 만든 후 실행한다.

1. make
2. ./main

 <div> <p>국민대학교</p> <p>소프트웨어부</p> <p>C++프로그래밍</p> </div>	결과보고서		
	프로젝트 명		
	팀 명		
	스네이크	Version 1.3	2022-JUN-15

5.3 실행 화면 캡처

