



국민대학교
소프트웨어융합대학
소프트웨어학부

C++프로그래밍 프로젝트

프로젝트 명	SnakeGame
팀 명	C++구조대 (9조)
문서 제목	최종보고서_SnakeGame_CPP구조대

Version	1.1
Date	2024-JUN-16

팀원	이 재원 (팀장)
	조 시현
	조 윤주
	함 태원

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-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 "**C++구조대**"의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 "**C++구조대**"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역

Filename	최종보고서_SNAKEGAME_CPP구조대.docx
원안작성자	이재원, 조시현, 조윤주, 함태원
수정작업자	이재원, 조시현, 조윤주, 함태원

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2024-05-20	이재원	1.0	최초 작성	
2024-06-16	이재원	1.1	수정	자기평가 작성


 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

목 차

1	개요	5
1.1	구조 및 개발 내용	5
	5
2	개발 내용 및 결과물	6
2.1	목표	6
2.2	개발 내용 및 결과물	7
2.2.1	개발 내용	7
	0단계: Window, Point	7
	1단계: Map의 구현	7
	2단계: Snake 표현 및 조작	8
	3단계: Item 요소의 구현	8
	4단계: Gate 요소의 구현	8
	5단계: 점수 요소의 구현	9
	추가	9
2.2.2	시스템 구조 및 설계도	9
	0단계: Window, Point	9
	1단계: Map	11
	2단계: Snake	12
	3단계: Item, Fruit, Poison	13
	4단계: Gate	15
	5단계: Score, Mission	16
	추가: GameTimer	18
	총합: Main, GameProcess	19
2.2.3	활용/개발된 기술	21
2.2.4	현실적 제한 요소 및 그 해결 방안	22
2.2.5	결과물 목록	22
3	자기평가	25
4	참고 문헌	27
5	부록	27
5.1	사용자 매뉴얼	27
5.1.1	Game Rule	27
5.1.2	Snake	28
5.1.3	Item	28

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

5.1.4	Gate.....	28
5.1.5	Mission.....	28
5.2	설치 방법.....	29

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

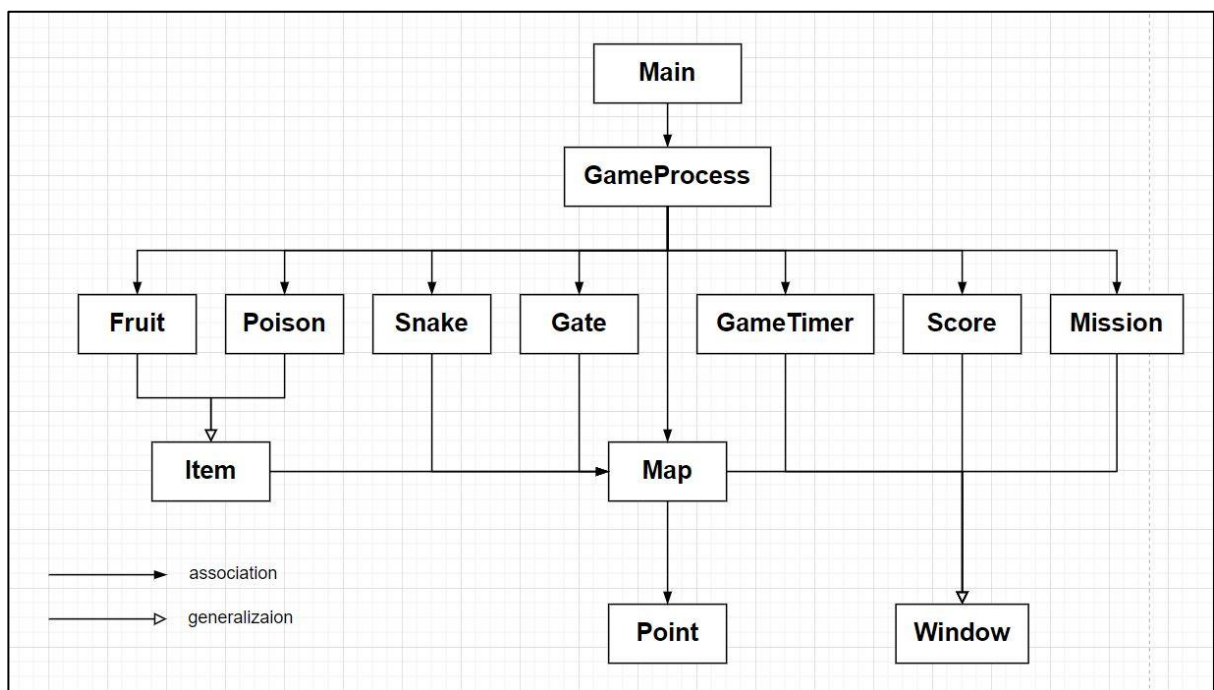
1 개요

평가기준 (10점)

프로젝트를 완성하기 위해 사용한 개발 방법을 기술하세요.

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

1.1 구조 및 개발 내용



개발을 진행하며 UML 다이어그램을 설계를 동시에 진행하여 구조적인 설계를 추구했다.


해당 프로젝트는 C++ 프로그래밍 언어와 표준 라이브러리, 외부 라이브러리를 통해 개발했다.

▷ 표준 라이브러리: fstream, random

- fstream: Map의 2차원 배열을 txt파일로 받기 위해 사용했다.
- random: Gate와 Item(Fruit, Poison)의 무작위 위치를 선정하기 위한 난수 발생을 위해 사용했다.
- queue: Snake의 움직임을 표현하기 위해 사용했다.

▷ 외부 라이브러리: ncurses, make

- ncurses: ncurses (new curses)는 프로그래머가 텍스트 사용자 인터페이스를 터미널 독립 방식으로 기록할 수 있도록 API를 제공하는 프로그래밍 라이브러리이다.

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

- make: 파일 간의 종속 관계를 파악하여 makefile에 적힌 대로 컴파일러에 명령하여 셀이 순차적으로 실행하게 해주는 라이브러리이다.


2 개발 내용 및 결과물

2.1 목표

작성요령 (10점)

프로젝트의 목표를 기술하세요. 각 단계별 목표를 구체적으로 쓰세요.

적용단계	내용	적용 여부
0단계	Window, Point	적용
1단계	Map의 구현	적용
	txt 파일로 2차원 배열 입력	적용
	2차원 배열 화면 출력	적용
	2차원 배열 값 반환/변경	적용
2단계	Snake 표현 및 조작	적용
	Queue로 Snake의 head, tail 구현	적용
	입력 받은 방향으로 nextHead 구현	적용
3단계	Item 요소의 구현	적용
	Map의 2차원 배열 접근	적용
	랜덤한 위치에 Item 생성	적용
	Snake가 Item 획득, 혹은 일정 틱 이후 Item 재 생성	적용
	Snake가 Item 획득 시 Snake의 길이 증가/감소	적용
4단계	Gate 요소의 구현	적용
	Map의 2차원 배열 접근	적용
	랜덤한 위치에 Gate 생성	적용
	Snake가 Gate 진입, 혹은 일정 틱 이후 Gate 재생성	적용
	Snake가 Gate 진입 시 진출 Gate 방향 알고리즘 구현	적용
5단계	점수 요소의 구현	적용
	Score 및 Mission 구현 및 출력	적용
	Mission 달성 시 게임 클리어 구현	적용
추가	GameTime 인터페이스 구현	적용
	GameOver/GameClear 화면 구현	적용
	Makefile	적용

 <div> 국민대학교 소프트웨어융합대학 C++프로그래밍 </div>	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-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.

0단계: Window, Point


- 화면에 출력될 요소들(Map, GameTimer, Score, Mission)은 모두 ncurses의 요소들을 사용하기 때문에 동일한 기능을 하는 함수들은 묶어서 상속하는 것이 좋다고 판단했다. 그렇기에 Window 클래스를 생성하여 Window를 부모클래스로 하여 개발을 진행했다.
- Map의 2차원 배열에 접근할 때 (x, y) 두 개의 변수를 사용한다. 이때 용이한 관리를 위해 Point 클래스를 생성하고 Point에서 (x, y) 두 가지를 모두 접근할 수 있도록 구현했다.

1단계: Map의 구현

- txt 파일로 2차원 배열 입력: 미리 txt 파일에 Map의 데이터를 저장하고 이를 파일 입출력을 통해 받아와 2차원 배열 형식으로 저장한다. 만일 다음 스테이지로 넘어갈 경우 다음 스테이지에 해당하는 txt 파일을 다시 가져와 2차원 배열에 저장한다.
- 2차원 배열 화면 출력: 2차원 배열에 저장된 값에 알맞은 값을 터미널에 출력한다.

2차원 배열 값	터미널 출력 값	설명
'0'	' '	Blank
'1'	'#'	Immune Wall
'2'	'#'	Wall
'3'	'H'	Snake Head
'4'	'B'	Snake Body
'5'	'F'	Fruit Item
'6'	'P'	Poison Item
'7'	'G'	Gate

- 2차원 배열 값 반환/변경: 다른 요소들이 2차원 배열의 값을 필요로 할 때 반환하고 필요시 2차원 배열의 값을 변경한다.

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

2단계: Snake 표현 및 조작

- Queue로 Snake의 head, tail 구현: Snake의 움직임을 표현하기 위해 Queue를 이용하여 구현한다. 먼저 Queue의 front는 tail, back은 head로 구현하였다. Snake의 head에서 현재 방향을 통해 nextHead를 구하고, 이를 Queue에 push한다. 그리고 Snake가 줄어들 때에는 pop으로 tail이 지워질 수 있도록 한다.
- 입력 받은 방향으로 nextHead 구현: 사용자의 키보드 입력에 따라 Snake의 방향을 전환하기 위해 ncurses의 wgetch()를 이용하여 구현한다. 이때 현재 방향에서 180도 회전한 방향이 입력될 경우 GameOver가 되도록 구현한다.

3단계: Item 요소의 구현

- Map의 2차원 배열 접근: Item 요소들의 좌표를 설정하기 위해 Map의 2차원 배열에 접근할 수 있도록 한다. 접근자는 getMap(), 설정자는 setMap()으로 구현했다. 이때 함수 인자로 int y, int x 2개의 값이 들어오거나 Point 객체 1개의 값이 들어올 수 있는 상황을 모두 상정하여 Function Overloading을 통해 이를 구현했다.
- 랜덤한 위치에 Item 생성: 정해진 조건에 따라 Item이 생성될 수 있는 위치 중에 random을 통해 생성된 난수를 이용하여 랜덤한 Item의 위치를 정해 생성한다.
- Snake가 Item 획득, 혹은 일정 틱 이후 Item 재생성: Snake가 Item을 획득하거나, 생성 후 일정 틱이 지났을 경우 현재 Item을 삭제하고 새로운 위치에 Item을 재 생성한다.
- Snake가 Item 획득 시 Snake의 길이 증가/감소: Snake가 Fruit을 획득하면 Snake 길이가 증가하고 Poison을 획득하면 길이가 감소하는 로직을 구현한다.

4단계: Gate 요소의 구현

- Map의 2차원 배열 접근: Gate 요소들의 좌표를 설정하기 위해 Map의 2차원 배열에 접근할 수 있도록 한다. Gate 클래스 안에서 Map 객체를 만들고, Gate 객체가 생성 될 때 함수 인자로 먼저 생성된 Map 객체의 포인터를 받아 이를 구현했다.
- 랜덤한 위치에 Gate 생성: 정해진 조건에 따라 Gate가 생성될 수 있는 위치 중에 random을 통해 생성된 난수를 이용하여 랜덤한 Gate의 위치를 정해 생성한다.
- Snake가 Gate 진입, 혹은 일정 틱 이후 Gate 재생성: Snake가 Gate에 진입하거나, 생성 후 일정 틱이 지났을 경우 현재 Gate를 삭제하고 새로운 위치에 Gate를 재 생성한다.
- Snake가 Gate 진입 시 진출 Gate 방향 알고리즘 구현: Gate 진입시의 방향과 진출 될 Gate의 상하좌우 값들을 확인하여 진출 시 Gate의 방향을 정하는 알고리즘을 구현한다. 먼저 직진이 가능한지, 다음으로 시계방향으로 90도 회전이 가능한지, 다음으로 반시계방향으로 90도 가능한지, 마지막으로 180도 회전이 가능한지를 체크한다.

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

5단계: 점수 요소의 구현

- Score 및 Mission 구현 및 출력: Game의 상태를 나타내주는 Score와 Mission을 구현한다. Map에서 행해지는 상호작용으로 발생하는 변화를 즉각적으로 반영하여 Score와 Mission에서 출력되도록 구현했다
- Mission 달성 시 게임 클리어 구현: Mission의 각 요소들의 각 Mission들이 Clear 되었을 경우 next stage로 넘어가거나, last stage일 경우 종료되는 로직을 구현한다.

추가

- GameTime 인터페이스 구현: 현재 진행중인 Game이 얼마나 진행되었는지 나타내주는 Gametime을 구현한다.
- GameOver/GameClear 화면 구현: Game이 실패할 경우, last stage가 아닐 때 clear한 경우, last stage에서 clear한 경우, 이렇게 총 3가지 경우에 해당할 때 각 화면을 출력한다.
- Makefile : 진행중인 SnakeGame프로젝트는 여러 클래스와 소스코드 파일로 나뉘어져 있기 때문에 컴파일 용이성을 위해 makefile을 작성하고 make를 이용해 컴파일했다.

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.

0단계: Window, Point

Window 클래스

- 개발자: 이재원
- 수정: 이재원
- 소스파일: Window.hpp, Window.cpp
- attribute

access specifiers	type	identifier	description
protected	Window*	win	Window 클래스를 상속받을 자식클래스들이 각각 제어할 윈도우 객체

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

- method

access specifiers	return	identifier	description
public	void	startColor()	ncurses의 start_color() 함수를 멤버 함수로 정의
public	void	endWin()	ncurses의 endwin() 함수를 멤버 함수로 정의
public	void	clear()	ncurses의 wclear() 함수를 멤버 함수로 정의
public	void	addCh(int y, int x, char ch)	mvwaddch()로 win에 정해진 위치에 한 글자 출력하는 함수
public	void	addStr(int y, int x, const char *str)	mvwaddStr()로 win에 정해진 위치에 여러 글자 출력하는 함수
public	void	refresh()	ncurses의 refresh() 함수를 멤버 함수로 정의

Point 클래스

- 개발자: 함태원
- 수정: 함태원, 이재원
- 소스파일: Point.hpp, Point.cpp
- attribute

access specifiers	type	identifier	description
protected	int	x, y	Point 클래스의 int형 x, y 변수

- method

access specifiers	return	identifier	description
public		Point(int y=0, int x=0)	생성자
public	Point	nextPoint()	방향 입력 후 다음 좌표를 반환
public	int	getX()	x 변수 접근자
public	int	getY()	y 변수 접근자
public	bool	operator==(const Point& nPoint)	Point 간의 == 연산자 중복

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

1단계: Map

Map 클래스

- 개발자: 함태원
- 수정: 함태원, 이재원
- 소스파일: Map.hpp, Map.cpp
- attribute

access specifiers	type	identifier	description
private	int	mapStage	현재 스테이지 변수
private	char	mapArray[MAP_SIZE][MAP_SIZE]	Map의 2차원 배열

- method

access specifiers	return	identifier	description
public		Map(const int height, const int width)	생성자
public	void	init()	Map 초기 설정
public	chtype	getInput()	Map의 win에서 입력받는 함수
public	char	getMap(const int y, const int x)	2차원 배열 접근자 매개변수: int, int
public	char	getMap(const Point point)	2차원 배열 접근자 매개변수: Point
public	void	setMap(const int y, const int x, const char c)	2차원 배열 설정자 매개변수: int, int, char
public	void	setMap(const Point point, const char ch)	2차원 배열 설정자 매개변수: Point, char
public	void	addCh(const int y, const int x, const char ch)	win에 (y, x)위치에 ch 출력
public	void	addStr(const int y, const int x, const char* ch)	win에 (y, x)위치에 *ch 출력
public	void	setMapFirst()	txt파일에서 입력 후 2차원 배열 저장
public	void	refreshMap()	win에 2차원 배열 전체 출력
public	bool	checkMap(const Point point, const char ch)	point위치에 ch가 있는지 검사
public	void	setTick(const int time)	Tick 설정자
public	void	checkLastStage()	현재 Stage가 last인지 확인

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

2단계: Snake

Snake 클래스

- 개발자: 함태원
- 수정: 함태원, 이재원
- 소스파일: Snake.hpp, Snake.cpp
- attribute

access specifiers	type	identifier	description
private	queue<Point>	snakeQueue	snake 움직임/좌표 역할 Queue
private	Direction	direction	방향 변수
private	Map*	thisMap	먼저 선언된 Map을 가리키는 변수
private	int	snakeLength	현재 snake의 길이 변수

- method

access specifiers	return	identifier	description
public		Snake(Map* map)	생성자
public	void	addPiece(Point piece)	Snake의 길이가 길어질 때 사용
public	void	removePiece()	Snake의 길이가 줄어들 때 사용
public	Point	tail()	현재 꼬리의 위치
public	Point	head()	현재 머리의 위치
public	Direction	getDirection()	현재 머리가 진행하고 있는 방향
public	bool	setDirection(Direction d)	방향 조정
public	void	inGateChangeDirection (Direction d)	win에 (y, x)위치에 ch 출력
public	Point	nextHead()	다음 head의 위치

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

3단계: Item, Fruit, Poison

Item 클래스

- 개발자: 조시현
- 수정: 조시현, 이재원
- 소스파일: Item.hpp, Item.cpp
- attribute

access specifiers	type	identifier	description
private	int	resetTime	Item 재생성 주기 변수
private	int	item	Item의 랜덤 위치
private	Point	point	Item의 위치
private	Map*	thisMap	먼저 선언된 Map을 가리키는 변수

- method

access specifiers	return	identifier	description
public		Item(Map* map)	생성자
public	Point&	getPoint()	Item 접근자
public	bool	makeItem(int& countSpace, const char ch)	item 변수에 랜덤값 설정 후 새로운 Item 생성 랜덤값: 0~countSpace-1
public	void	setItem(const int item, Point& point, int& countSpace, const char ch)	item번째의 '0'를 Item 설정 설정 후 countSpace -= 1 point변수에 설정한 Item 좌표 저장
public	void	deleteItem()	item 변수 초기화 point 위치의 Map 초기화
public	void	resetItem(int& countSpace, const char ch)	재생성 주기가 지났을 경우 deleteItem(), makeItem() 실행

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

Fruit 클래스


- 개발자: 조시현
- 수정: 조시현, 이재원
- 소스파일: Fruit.hpp, Fruit.cpp
- method

access specifiers	return	identifier	description
public		Fruit(Map* map)	생성자

Poison 클래스

- 개발자: 조시현
- 수정: 조시현, 이재원
- 소스파일: Poison.hpp, Poison.cpp
- method

access specifiers	return	identifier	description
public		Poison(Map* map)	생성자

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

4단계: Gate


Gate 클래스

- 개발자: 조시현
- 수정: 조시현, 이재원
- 소스파일: Gate.hpp, Gate.cpp
- attribute

access specifiers	type	identifier	description
private	int	resetTime	Gate 재생성 주기 변수
private	int	countWall	Wall의 총 개수
private	int	gate1	첫 번째 Gate 순서 위치
private	int	gate2	두 번째 Gate 순서 위치
private	Point	gate1Point	첫 번째 Gate 위치
private	Point	gate2Point	두 번째 Gate 위치
private	Map*	thisMap	먼저 선언된 Map을 가리키는 변수

- method

access specifiers	return	identifier	description
public		Gate(Map* map)	생성자
public	void	makeGate()	gate1 변수에 랜덤값 설정 후 gate2 변수에 gate1 변수와 다른 랜덤값 설정 새로운 Gate 생성 랜덤값: 0~countWall-1
public	void	setGate(const int gate, Point &gatePoint)	gate1번째의 '1'를 Gate1설정 gate2번째의 '1'를 Gate2설정 gate1Point/gate2point 변수에 설정된 gate 좌표 저장
public	Direction	inGate(Point& gatePoint, Direction diretion)	snake가 gate에 들어온 함수 순서대로 직진, 시계방향 90도, 반시계방향 90도, 180도 위치확인
public	bool	checkRange(Point point)	point가 범위 내 좌표인지 확인
public	int	getCountWall()	Wall의 개수 확인 및 저장
public	void	deleteGate()	생성된 게이트 제거
public	void	resetGate()	재생성 주기가 지났을 경우 deleteGate(), makeGate() 실행

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

5단계: Score, Mission


Score 클래스

- 개발자: 조윤주
- 수정: 조윤주, 이재원
- 소스파일: Score.hpp, Score.cpp
- attribute

access specifiers	type	identifier	description
private	int	currentLength	snake 현재 길이
private	int	maxLength	snake 최대 길이
private	int	growthItems	fruit 획득 횟수
private	int	poisonItems	poison 획득 횟수
private	int	usedGates	gate 이용 횟수

- method

access specifiers	return	identifier	description
public		Score(int height, int width, int y, int x)	생성자
public	int	getMaxLength()	maxLength 접근자
public	int	getGrowthItems();	growthItems 접근자
public	int	getPoisonItems();	poisonItems 접근자
public	int	getUsedGates();	usedGates 접근자
public	void	init()	테두리 출력
public	void	updateLength(const char ch)	ch == '+' 이면 updateGrowth() ch == '-' 이면 updatePoison() 그 후 updateMaxLen() 실행
public	void	updateMaxLen(const int maxLen)	입력된 maxLen이 저장된 maxLength보다 클 경우 저장
public	void	updateGrowth()	currentLength 1 증가 growthItems 1 증가
public	void	updatePoison();	currentLength 1 감소 poisonItems 1 증가
public	void	updateUseGate();	usedGates 1 증가
public	void	printBoard()	모든 요소 값 출력

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

Mission 클래스

- 개발자: 조윤주
- 수정: 조윤주, 이재원
- 소스파일: Mission.hpp, Mission.cpp
- attribute

access specifiers	type	identifier	description
private	int	missionLength	목표 snake 길이
private	int	missionGrowth	목표 fruit 획득 횟수
private	int	missionPoison	목표 Poison 획득 횟수
private	int	missionGate	목표 gate 이용 횟수
private	bool	boolLength	목표 length 달성 여부
private	bool	boolGrowth	목표 growth 달성 여부
private	bool	boolPoison	목표 poison 달성 여부
private	bool	boolGate	목표 gate 달성 여부

- method

access specifiers	return	identifier	description
public		Mission(int height, int width, int y, int x)	생성자
public	void	init()	테두리 출력
public	void	checkLength(const int maxLength)	목표 length 달성 여부 달성 시 v 출력
public	void	checkGrowth(const int growthItems)	목표 growth 달성 여부 달성 시 v 출력
public	void	checkPoison(const int poisonItems)	목표 poison 달성 여부 달성 시 v 출력
public	void	checkUsedGates(const int usedGates)	목표 gate 달성 여부 달성 시 v 출력
public	bool	checkMission()	모든 Mission 달성 여부 확인
public	void	printBoard()	모든 요소 값 출력

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

추가: GameTimer

GameTimer 클래스

- 개발자: 조윤주
- 수정: 조윤주, 이재원
- 소스파일: GameTimer.hpp, GameTimer.cpp
- attribute

access specifiers	type	identifier	description
private	int	gameTime	현재 game 진행 시간

- method

access specifiers	return	identifier	description
public		GameTimer(int height, int width, int y, int x)	생성자
public	void	initPrint()	테두리 출력 및 printBoard() 실행
public	void	printBoard()	gameTime 출력
public	void	upGameTime()	gateTime 증가 설정자

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

총합: Main, GameProcess

Main


- 개발자: 이재원
- 수정: 이재원
- 소스파일: Main.cpp
- attribute

access specifiers	type	identifier	description
—	int	main()	snakeGame 실행

GameProcess 클래스


- 개발자: 이재원
- 수정: 이재원
- 소스파일: GameProcess.hpp, GameProcess.cpp
- attribute

access specifiers	type	identifier	description
private	int	snakeLength	snake 현재 길이
private	Map*	map	Map 객체
private	GameTimer*	gameTimer	GameTimer 객체
private	Score*	score	Score 객체
private	Mission*	mission	Mission 객체
private	Snake*	snake	Snake 객체
private	Fruit*	fruit[5]	Fruit 객체 5개
private	Poison*	poison[5]	Poison 객체 5개
private	Gate*	gate	Gate 객체
private	int	gateInSnake	snake가 gate통과중일 때 기다려야 할 변수
private	bool	inGating	snake가 gate통과중인지 확인
private	bool	gaveOver	gameOver 확인 변수
private	int	countSpace	2차원 배열의 '0' 개수
private	bool	gameClearState	현재 stage의 clear상태

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

- method

access specifiers	return	identifier	description
public		GameProcess(const int height, const int width)	생성자
public	void	init()	모든 멤버 변수 초기화
public	void	initNextStage()	nextStage로 변환
public	void	deleteGame()	nextStage로 변환 전 기존 요소 삭제
public	void	initSnake()	초기 snake 초기화
public	void	initCountSpace()	Map 2차원 배열의 '0' 개수 확인
public	void	reduceSnake()	snake 길이가 감소 시 실행
public	void	inputGame()	사용자의 키보드 입력을 받는 함수 현재 방향과 180도 반대인 방향 시 Gameover
public	void	updateGame()	여러 요소들의 다음 행동 실행
public	void	reDraw()	Map의 재출력
public	void	setTick(const int time)	Tick 설정자
public	bool	checkFailed(const Point& next)	다음 좌표가 Wall, Immune Wall, SnakeHead, SnakeBody면 Gameover
public	bool	checkFruit(const Point& next)	다음 좌표가 Fruit이면 실행
public	bool	checkPoison(const Point& next)	다음 좌표가 Poison이면 실행
public	void	checkGate(Point& next)	다음 좌표가 Gate면 실행
public	void	checkTick()	재생성 주기가 있는 요소들의 Tick Check
public	void	checkMission()	현재 Mission 달성 여부 Check
public	bool	checkOver()	GameOver 면 finish() 실행
public	bool	isOver()	gameOver 접근자
public	bool	gameClear()	gameClearState 접근자
public	void	finish()	3가지 상태에 따라 출력 1- 마지막 스테이지에서 Clear 2- 마지막 스테이지가 아닌 스테이지에서 Clear 3- GameOver

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

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.

Ncurses

이번 SnakeGame 프로젝트의 중심이 된 라이브러리이다. Ncurses를 통해 사용자의 입력을 받고, 터미널에 출력한다. 특히 가장 상위 윈도우인 stdscr 아래로 Map, GameTimer, Score, Mission을 구현해 계층적인 형태의 프로그램 구조를 개발할 수 있었다. 또한 Ncurses의 Keypad 함수를 통해 방향키 입력과 같은 특수키의 입력을 수월하게 받아낼 수 있었다. 출력 역시 터미널에서 유기적으로 움직이는 그래픽을 표현하기 수월했다.

Queue

Snake의 움직임을 표현하게 해준 Queue 라이브러리이다. STL 라이브러리인 Queue를 통해 명확하고 간결한 Snake 움직임을 표현할 수 있었다. 단순 이동시 방향을 통해 다음 head 위치를 받아

Queue에 push하고 이동 후 pop을 통해 tail을 제거한다. 그리고 Fruit을 획득했을 때 pop 과정을 제외하면서 쉽게 구현할 수 있었다. 반대로 Poison을 획득했을 때 역시 pop 과정을 2번 거치며 구현할 수 있었다.

Make

make는 프로젝트 빌드에 필요한 명령들을 makefile이라는 파일에 기술하여, 파일의 내용에 따라서 편리하게 프로젝트를 빌드할 수 있는 외부 유틸리티이다. make를 사용함으로써 반복적인 컴파일 명령들을 일괄적으로 실행함으로써 시간을 절약할 수 있었다. 또한 소스코드 내의 종속적인 관계를 빠르게 파악할 수 있어 프로그램 소스 관리에도 도움이 되었다.

Random을 이용한 Item/Gate 생성

표준 라이브러리인 Random을 통해 난수를 생성하여 Item과 Gate 생성을 구현했다. 기존 C의 srand와 rand의 경우 seed에 따라 난수 값이 정해지는 문제가 있어 C++ 표준 라이브러리인 Random을 이용했다. 먼저 Item의 경우 Map의 2차원 배열의 '0', 즉 Item이 생성될 수 있는 공간의 개수인 countSpace를 구한다. 그리고 Random을 이용해 0부터 countSpace-1까지의 난수를 생성하고 다시 해당 난수에 해당하는 2차원 배열의 '0'을 Item 값으로 변경하여 Item 생성을 완료한다. Gate도 마찬가지로 countWall의 개수-1까지의 난수를 구하고 Gate를 생성한다. 단, Gate는 두 개가 한 쌍이되기에 Gate1과 Gate2의 난수 중복을 제거하는 것이 중요하다.

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

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.

Map의 접근자와 설정자의 Function Overloading

Map을 구현하면서 한 가지 문제점이 발생했다. 접근자와 설정자를 사용할 때 int y, int x 두개의 값이 전달되는 경우가 있고, Point 객체 한 개가 전달되는 경우가 있었다.

이 때 접근자와 설정자 각각 매개변수가 int형 두개이거나 Point 객체 한 개가 전달되게 하도록 Function Overloading을 구현하여 해당 문제를 해결하였다.

Map의 2차원 배열 접근

프로젝트 개발을 진행하면서 구조적인 문제점이 발생했다. 다른 요소(Snake, Item, Gate)들이 Map의 2차원 배열에 대한 접근이 필요해진 것이다. 하지만 GameProcess 클래스의 모두 같은 멤버 변수들이기에 구현하는 것이 어려웠다. 이 때 여기서 Snake, Item, Gate들의 생성자에 매개변수로 Map을 넘겨주면서 각각 요소들이 Map* 을 멤버변수로 하면서 이를 해결하였다.

2.2.5 결과물 목록

작성요령 (5점)

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

(1) 프로젝트 진행, 결과 및 동작에 관한 동영상 제작

- 동영상을 제작하고
- youtube에 upload한 후에, youtube 링크를 리스트할 것.

파일명	역할
map/map1~4.txt	Map의 2차원 배열에 저장될 초기 맵 데이터 값
Fruit.cpp	Fruit.hpp에서 정의한 메소드를 구현하는 소스코드
Fruit.hpp	Snake가 먹으면 길이가 1 증가하는 Item을 정의한 소스코드
GameProcess.cpp	GameProcess.hpp 에서 정의한 메소드를 구현하는 소스코드
GameProcess.hpp	게임의 흐름을 제어하는 GameProcess 클래스 소스코드

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

GameTimer.cpp	GameTimer.hpp에서 정의한 메소드를 구현하는 소스코드
GameTimer.hpp	현재 스테이지의 Tick이 얼마나 지났는지 구현하는 소스코드
Gate.cpp	Gate.hpp에서 정의한 메소드를 구현하는 소스코드
Gate.hpp	Gate의 생성, 삭제, 제어를 관리하는 소스코드
Item.cpp	Item.hpp에서 정의한 메소드를 구현하는 소스코드
Item.hpp	Fruit, Poison의 부모 클래스 소스코드
Main.cpp	SnakeGame을 실행시키는 메인 소스코드
Map.cpp	Map.hpp에서 정의한 메소드를 구현하는 소스코드
Map.hpp	SnakeGame의 요소들이 저장되는 Map의 생성, 삭제, 제어를 관리하는 소스코드
Mission.cpp	Mission.hpp에서 정의한 메소드를 구현하는 소스코드
Mission.hpp	각 스테이지의 수행될 Mission을 제어하고 출력하는 소스코드
Point.cpp	Point.hpp에서 정의한 메소드를 구현하는 소스코드
Point.hpp	x, y의 값을 하나의 클래스로 구현하는 소스코드
Poison.cpp	Poison.hpp에서 정의한 메소드를 구현하는 소스코드
Poison.hpp	Snake가 먹으면 길이가 1 감소하는 Item을 정의한 소스코드
Score.cpp	Score.hpp에서 정의한 메소드를 구현하는 소스코드
Score.hpp	현재 스테이지에서 수행한 것들을 제어하고 출력하는 소스코드
Snake.cpp	Snake.hpp에서 정의한 메소드를 구현하는 소스코드
Snake.hpp	Snake의 생성, 움직임, 증가, 감소를 구현하는 소스코드
Window.cpp	Window.hpp에서 정의한 메소드를 구현하는 소스코드
Window.hpp	Map, GameTimer, Mission, Score 클래스의 부모 클래스 소스코드

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

시연영상

<https://youtu.be/H4i0L1Qh7f8>

이재원: Main, GameProcess, Window

<https://youtu.be/S-cB9fhF8CQ>

조시현: Item, Fruit, Poison, Gate


<https://youtu.be/oV7l21mjm1o>

조윤주: GameTimer, Mission, Score

<https://youtu.be/8MpKbPL6kNo>

함태원: Point, Snake, Map

<https://youtu.be/3GIJezr2HI8>


 <div> 국민대학교 소프트웨어융합대학 C++프로그래밍 </div>	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

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.

이재원	<p>팀장으로서 이번 프로젝트를 주도하고, 전체적인 SnakeGame의 흐름과 팀원들의 코드를 종합하는 역할을 맡았다.</p> <p>프로젝트를 진행하기 앞서 한 가지 걱정이 있었다. 팀원 모두 C++이라는 새로운 언어로 각자의 파트를 나눠 진행하는 분할 적인 프로젝트가 생소하다는 것이었다. 나 역시도 이렇게 세부적으로 역할을 나눠 진행해보는 프로젝트는 처음이었기에 프로젝트를 원활하게 진행하기 위해서는 앞선 준비가 필요하다고 생각했다. 무작정 각자 개발을 시작하는 것이 아니라 하나의 규칙을 정하고 따르며 코드를 짜고 개발하는 것이 좋을 것이라 판단했다. 그렇게 우리 팀은 개발에 앞서 먼저 code template을 정했다. 사람마다 코드를 작성하는 스타일이 다르기에 내가 보기엔 쉽게 해석할 수 있을지라도 남에게는 어렵게 보일 수 있다. 그래서 소스코드를 작성할 때의 명칭, 주석, 문장 규칙을 정해 모두가 통일화 할 수 있도록 했다. 처음에는 규제적인 코드작성이 불편했지만 팀원과의 코드를 서로 공유할 때 확실히 큰 도움이 됐다. 협업 과정에서 규칙을 정하고 획일화된 협업이 얼마나 중요한지 깨닫게 되었다.</p> <p>또한 협업을 위해 Github를 사용하였는데 이 점에서 아쉬움이 있었다. 처음에는 코드를 공유하고 사용할 때 좋았지만 개발과정에서 github와 연동되지 않았기에 git push를 적극적으로 사용하지 못했다. 그래서 github에 일일이 코드를 upload하는 식으로 업데이트했는데 git을 100% 사용하지 못한 것이 정말 아쉬웠다. 미래에 하게 될 프로젝트에서는 code template과 더불어 github 연동을 확실하게 해야겠다고 다짐했다.</p>
조시현	<p>본인은 이번 SnakeGame Project를 진행하면서 Item(Fruit, Poison) class의 제작을 맡았다. 또한, 아이디어 브레인스토밍 과정에서 SnakeGame 속 추가로 들어가면 좋을 아이디어 제언에 적극적으로 참여하였고, 게임 구상 틀과 자료 조사에도 성실히 참여하였다.</p> <p>프로젝트 제작 중 아직 C++언어를 배우는 과정에 있다보니, C++언어와 그 활용에 완벽히 익숙하지 않다는 점이 큰 난관이었다. 그러나, 이번 기회를 통해 직접 C++언어를 활용하여 프로젝트를 진행하였다는 점은 C++언어를 이해하는 과정에 큰 도움이 되었다. 프로젝트 제작 과정 중 제작에 필요한 언어에 익숙하지 않는다는 것은 큰 결점이기때문에, C++언어에 더 익숙한 상태였다면 결과가 더 좋지 않았을까 하는 아쉬움도 남는다.</p>

	국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서	
		프로젝트 명	SnakeGame
		팀 명	C++구조대
		Confidential Restricted	Version 1.1 2024-JUN-16

조윤주	<p>프로젝트 내에서 하위 창(sub window)과 GameTimer, Mission, Score 클래스의 구현을 맡았다. 프로젝트 초반에는 curses의 명령어가 기존 C++의 명령어와 차이가 있어 명령어를 익히는 데 어려움이 있었다. 이러한 어려움을 극복하기 위해 관련 문서를 정독하여 작동 원리를 이해하고, 실제 프로젝트에 적용하며 익숙해졌다. 팀원들과 소통하고 논의하며 진행 상황을 점검하는 과정에서 프로젝트의 진행 과정을 경험할 수 있었고, 클래스와 헤더 파일의 개념을 이해하는 데 도움이 되었다. 해당 프로젝트를 수행하며 사용해본 적 없던 라이브러리를 활용함으로써 많은 것을 배우고 성장할 수 있는 기회가 되었다. 미숙한 부분은 팀원들과 소통을 통해 피드백을 거듭하며 주제에 맞게 구현해냈고, 프로젝트에 적극적으로 참여했다.</p>
함태원	<p>게임에서 Snake와 Map을 맡았었다. 먼저 Snake게임이라고 해서 우리가 게임을 할 때는 간단하게 느껴져 프로그래밍도 간단할것이라고 생각했었는데 생각보다 생각해야하는 요소가 많았고 또다른 면으로는 C++로 하나부터 열까지 만들어볼 수 있다는 것에 설렘했다. 그리고 팀프로젝트로 진행하면서 내가 만든 부분과 팀원들이 만든 부분을 합칠때 어떤 이유때문에 실행이 잘 안되는지 찾는게 힘들었지만 합쳐져서 실행이 됐을때 신기함과 뿌듯함을 느꼈다.</p> <p>본격적인 코딩을 시작하기 전에 팀원들과 변수명을 어떻게 사용할지 각 아이템이나 부분들의 값을 어떻게 잡을지 토론을 하면서 설정을 했음에도 불구하고 다른사람에게 내가 한 프로그래밍을 이해시킨다는 것이 얼마나 힘든지도 깨달았다. 이부분은 뭔가 실제로 본업에서도 이렇게 할것같아서 미리 체험해보는 기분도 들었다. 그리고 내가 생각한 아이디어를 팀원들에게 공유하고 또 팀원들의 아이디어를 들으면서 팀프로젝트의 장점을 느끼게 되었다.</p> <p>아쉬웠던 점은 Github를 남에게 공유를 하면서 실제로 사용해본 경험이 이번이 처음이다보니 파일을 올리고 업데이트 하는것도 어색해서 힘들었고 프로젝트를 진행하면서 모르는 문법을 찾아보고 적용해보고 특히 ncurses에 관한 내용을 처음 접해보았기에 뭔가 빠격거린 부분이 스스로가 좀 아쉬웠다. 그렇지만 이번에 진행하면서 남들과 협업할때는 어떤점을 생각해야하는지에대해 경험하는 계기가 되어 좋았던것 같다.</p>

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

4 참고 문헌

참고한 서적, 기사, 기술 문서, 웹페이지를 나열한다.:

번호	종류	제목	출처	발행년도	저자	기타
1	웹사이트	[ncurses] 1. ncurses 개요 및 설치, Window의 개념	https://migmatart.dev/development/2017/06/02/ncurses1.html	2017	MagmaTart	
2	웹사이트	std::queue	https://cplusplus.com/reference/queue/queue/		cplusplus	
3	웹사이트	<17 - 3. 난수 생성(<random>)과 시간 관련 라이브러리(<chrono>) 소개>	https://modoocode.com/304	2020	모두의 코드	

5 부록

작성요령 (15점)


프로젝트의 결과물을 사용하기 위한 방법에 대해서 작성하세요.

5.1 사용자 매뉴얼

프로젝트 실행 후 사용안내, 따라하기 등 포함

5.1.1 Game Rule

- Snake는 0.5초(1틱)에 한 칸씩 움직인다.
- user는 Snake를 키보드의 방향키를 통해 상하좌우로 움직일 수 있다.
- 벽에 닿을 경우 게임은 끝난다.
- Fruit을 먹을 시 Snake가 늘어나고, Poison을 먹을 시 Snake가 줄어든다.
- 총 4개의 스테이지로 이루어져있다.

 국민대학교 소프트웨어융합대학 C++프로그래밍	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

5.1.2 Snake

- 'H' : Snake Head / 'B' : Snake Body
- 키보드의 방향키를 통해 입력하면 Snake Head의 방향이 변경된다.
- 단, 현재 Snake의 방향과 180도(예: 상->하, 우->좌) 반대의 방향을 입력할 경우 GameOver된다.
- Snake Head가 Wall, Immune Wall, Snake Body에 닿을 경우 GameOver된다.

5.1.3 Item

- 'F' : Fruit / 'P' : Poison
- Item 종류로는 Fruit과 Poison이 있다.
- 두 Item은 매 스테이지마다 각각 5개씩 생성된다.
- 두 Item은 랜덤한 위치에 생성되며, 생성 10틱 후 다른 랜덤한 위치에 재생성된다.
- 두 Item 획득 즉시 다른 랜덤한 장소에 재생성된다.
- Fruit 획득 시 Snake의 길이가 1 증가한다.
- Poison 획득 시 Snake의 길이가 1 감소한다.

5.1.4 Gate

- 'G' : Gate
- Gate는 매 스테이지마다 두 개로 랜덤한 위치에 한 쌍씩 생성된다.
- 한 Gate에 진입할 시 다른 Gate로 진출된다.
- Gate가 생성되고 10틱 후 다른 랜덤한 위치에 재생성된다.
- Gate에 진입 중에는 Gate가 사라지지 않는다.

5.1.5 Mission

- Mission은 총 4가지가 있다.
- 'B' : (10) – Snake의 최대 길이 10 달성
- '+' : (5) - Fruit 5번 획득하기
- '-' : (5) - Poison 5번 획득하기
- 'G' : (3) – Gate 3번 진입하기
- 각 미션 클리어 시 오른쪽 소괄호 안에 'v' 가 입력된다.
- 모든 미션을 클리어할 경우 다음 스테이지로 넘어간다.
- 마지막 스테이지에서 모든 미션을 클리어할 경우 게임이 종료된다.

 <div> 국민대학교 소프트웨어융합대학 C++프로그래밍 </div>	결과보고서		
	프로젝트 명	SnakeGame	
	팀 명	C++구조대	
	Confidential Restricted	Version 1.1	2024-JUN-16

5.2 설치 방법

프로그램 컴파일 및 실행 방법 포함

1. G++ 설치

~\$ sudo apt-get install g++

2. make 설치

~\$ sudo apt-get install make

3. Ncurses 설치

~\$ sudo apt-get install libncurses5-dev libncursesw5-dev

4. Git 설치

~\$ sudo apt-get install git

5. git clone

~\$ git clone https://github.com/2wodnjs7/2024-Cpp-Project.git

~\$ cd 2024-Cpp-Project/

~/2024-Cpp-Project\$ cd snake_ncurses/

~/2024-Cpp-Project/snake_ncurses\$ make

6. 실행

★ 실행 시 주의사항 ★

★ 터미널의 크기를 충분히 늘려 주시고 실행해주시길 바랍니다. ★

★ 터미널의 크기가 작을 경우 바로 GameOver가 될 수 있습니다. ★

~/2024-Cpp-Project/snake_ncurses\$./SnakeGameExe