

C++프로그래밍 프로젝트


프로젝트 명	Snake Game
팀 명	7조
문서 제목	결과보고서

Version	1.0
Date	2020-JUN-22

팀원	최혁태(팀장)
	윤민상
	김하연


CONFIDENTIALITY/SECURITY WARNING
<p>이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 C++프로그래밍 수강 학생 중 프로젝트 “Snake Game”를 수행하는 팀 “7조”의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 “7조”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.</p>

문서 정보 / 수정 내역

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22


Filename	최종보고서-SnakeGame.doc
원안작성자	최혁태, 윤민상, 김하연
수정작업자	최혁태, 윤민상, 김하연

수정날짜	대표수정 자	Revisio n	추가/수정 항목	내 용
2020-06-20	윤민상	1.0	최초 작성	최초 작성
2020-06-20	최혁태	1.1	내용 수정	수정된 연구내용 추가 / 코드 수정
2020-06-20	김하연	1.2	내용 수정	향후 추진 계획 수정 / 내용 흐름 수정

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

목 차

1	개요	4
2	개발 내용 및 결과물	5
2.1	목표	5
2.2	개발 내용 및 결과물	6
2.2.1	개발 내용	6
2.2.2	시스템 구조 및 설계도	6
2.2.3	활용/개발된 기술	6
2.2.4	현실적 제한 요소 및 그 해결 방안	6
2.2.5	결과물 목록	7
3	자기평가	8
4	참고 문헌	8
5	부록	8
5.1	사용자 매뉴얼	8
5.2	설치 방법	8

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

1 개요


평가기준 (10점)

프로젝트를 완성하기 위해 사용한 개발 방법을 기술하세요.

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

https://github.com/minsang22/CPP_Project_Team7 - 7조 프로젝트 GitHub 링크

개발 기간동안 매주 최소 한번 이상은 회의를 하며 프로젝트를 진행하였습니다. 회의는 카카오톡을 통해 미리 날짜를 협의한 후에 대면 회의 혹은 ZOOM의 화면 공유 기능을 활용하는 방식으로 진행하였고, 이전 회의에서 설정한 개발 계획을 어떤 방식으로 구현하였는지를 피드백하고 다음 회의까지 어느 부분을 누가 개발할지, 어떤 알고리즘을 사용할지에 대한 간략한 이야기를 나누었습니다. 각자 구현한 코드는 Slack을 사용해 실시간으로 공유하며 서로 코멘트를 통한 피드백을 진행하였고, Slack을 통해 프로젝트의 진행 상황 혹은 각종 정보를 공유하였습니다. 그리고 IOS의 공유메모 기능을 사용해 제작 일정을 구체적으로 기록해 가며 작업 효율을 높이도록 하였습니다. 프로젝트의 개발 단계를 기록하는 것은 하나의 개발 단계를 끝마칠때마다 GitHub에 소스 코드들을 Push해 각각 단계마다의 최종 코드를 기록하는 방식으로 진행하였습니다. 개발 단계에서 사용한 대표적인 라이브러리는 크게 Vector, Ncurses, ctime 라이브러리가 있고 Google 검색을 통해 이 라이브러리들을 접해 설치하고 사용하게 되었습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

2 개발 내용 및 결과물


2.1 목표

작성요령 (10점)

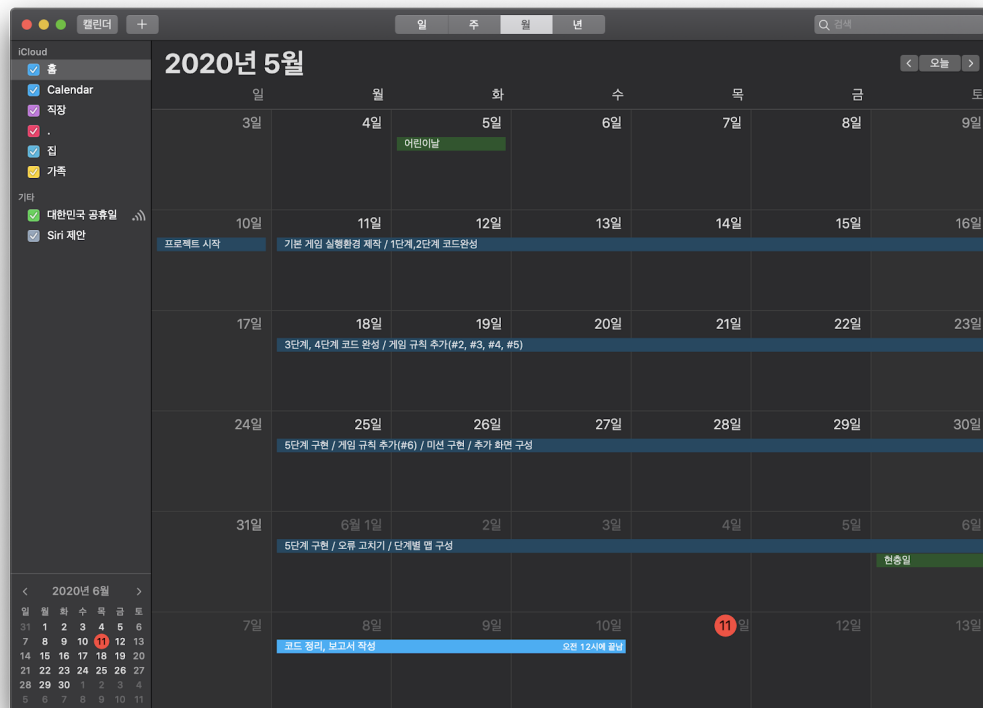
프로젝트의 목표를 기술하세요. 각 단계별 목표를 구체적으로 쓰세요.

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

- 1단계 : Snake Map의 구현
Ncurses 라이브러리의 기본 함수들을 사용해 Map과 점수, 미션을 표시하는 Window를 구현해 주었습니다.
- 2단계 : Snake와 Snake의 조작 구현
Snake가 움직이는 것을 구현함과 동시에 3단계에서 구현할 Growth Item을 미리 2단계에서 테스트해보았습니다. 그리고 게임 도중 일정 버튼을 통해 게임을 일시중지 하는 기능을 추가해주었습니다.
- 3단계 : Growth / Poison Item 구현
2단계에서 구현했던 Growth Item과 더불어 Poison Item 까지 구현해 주었습니다. 2단계와 달리 5초간 Item을 획득하지 못할시에 Item이 재생성되는 기능까지 추가해주었습니다.
- 4단계 : Gate 요소의 구현
Gate가 생성되는 것과 Gate를 올바른 방향으로 통과하고 재생성되는 과정까지 구현해주었습니다.
- 5단계 : 점수 요소의 구현
Map의 오른쪽에 Score Board와 Mission Board를 출력해 줄 수 있도록 구현해 주었습니다. 그리고 Mission을 클리어 함에 따라서 다음 단계의 Map으로 바뀌는 기능과 더불어 다양한 추가 기능을 이 단계에서 구현해 Snake Game을 완성했습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

- 실제 소요된 제작 일정




저희는 매주 단계별 코드를 짜고 게임의 규칙을 몇개씩 추가하는 방향으로 프로젝트를 진행했습니다.

1주차: Ncurses를 사용하여 터미널위에 맵을 출력하고 전체적인 틀이 되는 게임의 기본 실행 환경을 제작했습니다. 또한 1단계와 2단계 코드를 완성하여 뱀의 기본적인 움직임을 구현했습니다.

2주차: map 위에 Growth Item과 Poison Item을 출현하게 하는 3단계를 구현했고, map 안의 wall에서 한 쌍의 gate가 출현하게 하는 4단계까지 구현했습니다.

3주차: 맵 우측에 게임 점수를 표시하는 칸을 만들어 5단계를 달성했습니다.

4주차: 여러 오류를 수정하고, 단계별로 맵을 만들어 mission을 달성하면 단계가 올라가며 맵이 바뀌게 했습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

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.

1) Snake Map의 구현


주요 기능 : 2차원 배열로 Snake Map의 정보를 담아 Map을 표현하였습니다. Mission이 클리어됨에 따라 난이도가 올라가며 바꿀 Map을 총 5단계로 구성하였고, Map 전체를 감싸고 있는 부분은 3, Snake가 접촉할 수 없는 모서리 부분은 2, 일반 벽은 1, Snake가 이동할 수 있는 부분은 1로 표현한 후 숫자에 따라 구성인자들을 구분하는 방식으로 진행하였습니다.

수정한 소스 파일 : map.h, sclass.h

개발 클래스 / 함수 :

```
void mapChange(int next_map[30][60]){ // 맵 바꿔주기
    for(int i=0; i<30; i++){
        for(int j=0; j<60; j++){
            map[i][j] = next_map[i][j];
        }
    }
    checkWall(next_map);
}
```

- mapChange : 주어진 미션을 충족해 다음 단계로 넘어갈 때에 맵을 바꿔주기 위한 함수입니다. for 반복문을 돌며 map에 next_map을 복사해준 후, checkWall 함수를 호출해 벽의 위치를 저장하는 Vector 배열까지 만들어주도록 하였습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

```

char* checkMap(){ // 맵바꾸기
    if(level == 1){return getTable(map1);}
    if(level == 2){return getTable(map2);}
        if(level == 3){return getTable(map3);}
    if(level == 4){return getTable(map4);}
        if(level == 5){return getTable(map5);}
    return getTable(map5);
}

```

- checkMap : checkMap 함수를 통해 레벨에 따른 map을 return해주도록 하였습니다.


개발자 : 최혁태, 윤민상

2) Snake와 Snake의 조작 구현

주요 기능 : 좌표값을 통해 Snake에 각종 기능을 적용할 수 있도록 구현하고, 키보드로 입력값을 받아 Snake를 조작할 수 있도록 구현하였습니다.

수정한 소스 파일 :sclass.h / vector.h / snake3.cpp

개발 클래스 / 함수 :

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

```

class Vector
{
    public:
        Vector() : x(0), y(0){}
        Vector(int x, int y) {setX(x); setY(y);}

        void setX(int x){this->x = x;}
        int getX() const {return this->x;}
        void setY(int y){this->y = y;}
        int getY() const {return this->y;}

        Vector& operator=(const Vector& vec)
        {setX(vec.getX()); setY(vec.getY()); return *this;}
        Vector& operator+=(const Vector& vec)
        {setX(getX()+vec.getX()); setY(getY()+vec.getY()); return *this;}
        Vector operator+(const Vector& vec)
        {Vector v(getX(),getY()); return v+=vec; }
        Vector& operator-=(const Vector& vec)
        {setX(getX()-vec.getX()); setY(getY()-vec.getY()); return *this;}
        Vector operator-(const Vector& vec)
        {Vector v(getX(),getY()); return v-=vec;}
        bool operator==(const Vector& vec)
        {return getX()==vec.getX() && getY()==vec.getY();}
        bool operator!=(const Vector& vec)
        {return !operator==(vec);}

    private:
        int x,y;
};

```


- vector 클래스 : 강의에서 배운 연산자 재정의의 사용해 Snake의 조작과 각종 기능에서 좌표를 사용할 때 좌표값 사이의 연산을 편리하게 해주도록 하였습니다.

```

void getSize(){ //3이하로 뱀 크기 떨어지면 죽음
    if(body.size(<3){exit = true;}
}

```

- getSize : 게임의 기본 조건인 Snake의 size가 3보다 작아질 시 게임이 끝나게 되는 조건을 getSize 함수로 구현해주었습니다.

 <div> 국민대학교 컴퓨터공학부 캡스톤 디자인 I </div>	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

```

void setDirection(int d) // 방향 설정
{
    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;
    }
}

```

- setDirection : setDirection 함수는 Snake의 이동 방향을 설정해주기 위한 함수입니다. switch문을 통해 키보드로 입력받는 값에 따라 이동할 좌표값을 Vector로 설정해주도록 하였습니다.

```

//맵 다음단계 되면 다시3개로 줄여줌
void changeBody(){

    for(int i=0; i<body.size()-3; i++){
        reduceBodySize(body.size());
    }

}

```

- changeBody : 다음 단계의 Map으로 넘어갈 때에 Snake의 size를 다시 3으로 줄여주기 위해 만들어준 함수입니다. for 반복문을 돌며 Snake의 size가 3이 될 때까지 size를 감소시켜줍니다.


```

void makeMove() // 이동
{
    if(!exit)body[0]+=direction;

    for(unsigned int i = 1; i<body.size(); ++i)
        if(body[i]==body[0]){exit = true; break;}
}

void bodyMove()
{
    for(unsigned int i=body.size()-1; i>0; --i)
        increaseBodySize(i);
}

```

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

- makeMove, bodyMove : 실제로 Snake를 움직이게 해주는 함수입니다. 미리 선언해준 exit 변수가 true가 된다면 Snake가 Map 바깥으로 탈출했다는 의미이므로 게임이 끝나게 됩니다.
- procesInput : 키보드로 받는 입력을 Snake Game에 맞게 전달해주기 위한 함수입니다. 기본적으로 방향 키를 통해 Snake를 움직일 수 있도록 해주었고, p 버튼을 누를시 게임을 일시중지 하는 기능도 추가하였습니다.

개발자 : 김하연

3) Growth / Poison Item 구현

주요 기능 : 획득하였을 때 Snake의 Size가 1만큼 감소하거나 증가하게 해주는 Growth / Poison Item을 구현하였고, 5초동안 Item을 획득하지 못할 시에는 Item을 재생성해주도록 하였습니다.


수정한 소스 파일 : sclass.h

개발 클래스 / 함수 :

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

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

- dropApple, dropPoison : 랜덤 좌표를 정해준 뒤 그 좌표에 아이템을 생성해주는 함수입니다. 5초동안 Item을 획득하지 못할 시 Item이 재생성되는 기능을 추가하기 위해 이 함수가 호출될 때마다 현재 시간으로 startTime을 갱신하도록 하였고, 현재 시간을 갱신해줄때는 ctime 라이브러리의 time(NULL) 함수를 이용하였습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

```


void checkForApple()
{
    /*아이템을 먹을때마다 startT 갱신해줌.
    메인 while문에서 계속 checkForApple을 호출해주며
    nowT를 갱신해줌. 만약 nowT - startT가 5초 이상이
    될 경우 dropApple함수를 재호출해서 새로 아이템 생성*/

    nowAT = time(NULL);

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

```

- checkForApple, checkForPoison : snake3.cpp 파일의 main문 속 while문에서 Snake가 아이템을 획득했는지 좌표값을 통해 검사해줄 함수입니다. 반복문에서 호출되며 계속 nowTime을 현재 시간으로 갱신해주고, 만약 nowTime - startTime가 itemTime(== 5초)보다 커질 시엔 아이템을 재생성 해주는 식으로 기능을 구현해주었습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

```

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 : Item을 획득하였을 시에 Snake의 size를 효율적으로 조절하기 위해 구현한 함수입니다.

개발자 : 김하연, 윤민상

4) Gate 요소의 구현

주요 기능 : 벽의 좌표들중 랜덤 난수 생성을 통해 정해진 좌표에 양 쪽 Gate를 생성해 주었고, Gate를 올바른 방향으로 통과하는 동시에 재생성되는 과정까지 구현하였습니다.

수정한 소스 파일 : sclass.h


개발 클래스 / 함수 :

```

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

```

- checkWall : 벽의 위치를 저장해주기 위해 벽이 생성되는 위치를 Vector 배열로 생성해주는 함수입니다. for 반복문을 통해 Map 전체를 검사해주며 값이 1인

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

좌표(벽을 1로, 모서리를 2로 표현)들을 Vector 배열에 저장해주는 방식으로 구현하였습니다.


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

- getGate : 랜덤 난수 생성 기능을 통해 Gate의 양쪽 입구 좌표를 선택해주었습니다. 랜덤 난수는 checkWall 함수를 통해 Vector 배열에 저장해 놓은 값들 중에서만 생성하게 하여 벽에만 Gate가 생성되도록 구현하였습니다.

```
void checkGate() // 몸이 다 통과할때까지 머리 방향 설정
{
    if(body[0] == gate[0]){
        body[0] = gate[1];
        countG++;
        setDirection(gateDirection(gate[1]));
    }

    else if(body[0] == gate[1]){
        countG++;
        body[0] = gate[0];
        setDirection(gateDirection(gate[0]));
    }
}
```

- checkGate : Snake의 몸이 다 통과할 때 까지의 몸의 방향을 설정해주어, 한쪽 Gate를 통과하면 다른 쪽의 Gate로 나올 수 있도록 해주는 함수입니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22


```

void changeGate() // 다 통과해야 게이트 바뀌도록
{
    Vector A = direction;
    for(int i=0; i<body.size(); i++){
        A = A + direction;
    }
    if(body[0] == gate[0]+A){
        getGate(wall1);
    }

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

```

- changeGate : Snake의 몸이 Gate를 온전히 다 통과해야 다른 위치에 새로운 Gate를 재생성하도록 해 주는 함수입니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

```

char* getTable(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[apple.getY()*width+apple.getX()] = 'a';
    table[potion.getY()*width+potion.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;
}

```

- getTable : 생성한 1차원 배열에 맵의 정보와 Apple / Poison 아이템, Gate의 좌표와같은 모든 정보를 지정한 문자로 정해진 좌표에 표현해서 전체 배열을 사용하기 용이한 형태로 저장할 수 있도록 getTable 함수를 구현하였습니다.

개발자 : 김하연, 최혁태


5) 점수 요소의 구현

주요 기능 : 현재의 점수와 각 단계의 목표 Mission, 그리고 현재의 Mission 진행 상황을 출력해 주도록 하였습니다.

수정한 소스 파일 : snake3.cpp , sclass.h

개발 클래스 / 함수 :

- printScore, printMission : Ncurses 라이브러리의 mvwprintw 함수를 이용하여 1단계에서 사용했던 방식처럼 WINDOW 객체에 점수와 Mission 화면을 출력해 주도록 하였습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22


- mission : 게임 플레이 도중 Map의 오른쪽에 출력되는 Mission Board의 가독성을 높이기 위해 각 Mission의 조건을 충족할 시 v 표시를 해주는 기능을 추가해주는 함수를 구현하였습니다.

```
int missionComplete(){
    if(comP == 'v' && comA == 'v' && comG == 'v' && comB == 'v' ) {
        level ++;
        return level;
    }
    return level;
}
```

- missionComplete : 모든 Mission을 충족할 시에 게임이 다음 단계로 넘어갈 수 있도록 해주는 기능을 구현한 함수입니다.

```
bool allComplete(){
    if(level > 5){
        exit = true;
        return true;
    }
    return false;
}
```

- allComplete : 만약 모든 Mission을 만족해 게임을 클리어 했다면 게임을 종료해주는 기능을 구현한 함수입니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22


```

void checkPoints() // 미션 완료시 맵 바꾸고 미션 스코어 변경
{
    switch(missionComplete()){
        case 2: {
            missionA =2;
            missionB =4;
            missionP =2;
            missionG =2;
            mapChange(map2);
            break;}
        case 3: {
            missionA =3;
            missionB =5;
            missionP =3;
            missionG =3;
            mapChange(map3); break;}
        case 4: {
            missionA =4;
            missionB =6;
            missionP =4;
            missionG =4;
            mapChange(map4); break;}
        case 5: {
            missionA =5;
            missionB =7;
            missionP =5;
            missionG =5;
            mapChange(map5); break;}
    }
}

```

- checkPoints : Mission 완료 시 게임이 다음 단계로 넘어갈 때 mapChange 함수를 호출해 Map을 바꿔주는 동시에 새로운 Mission을 부여해주는 기능을 구현한 함수입니다.

개발자 : 윤민상, 김하연, 최혁태

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

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.

저희 프로그램에서 사용한 외부 라이브러리는 총 세개입니다.


Ncurses는 터미널에 맵과 스네이크, 아이템 등 여러 시각적인 요소들을 출력하기 위해 사용하였습니다.

Vector는 맵 위에 있는 모든 것(벽, 스네이크, 아이템)들의 x, y좌표값을 설정해주기 위해 사용하였습니다.

ctime은 5초가 지날 때마다 아이템의 위치를 바꿔주기 위함과, 게임이 시작할때부터 끝날때까지의 시간을 측정하여 게임 결과화면에서 걸린 시간을 출력해주기 위해 사용하였습니다.

또한 Makefile은 다음과 같이 작성합니다.

- **CC = g++** : C++ 컴파일러
 - **TARGET = SnakeGame** : 실행파일 이름
 - **SOURCES = SnakeGame.cpp** : 컴파일할 cpp 파일
 - **LD_FLAGS = -lncursesw** : 링커 옵션
-
- **all : \$(CC) -std=c++11 -o \$(TARGET) \$(SOURCES) \$(LD_FLAGS)**
 - **ex) g++ -std=c++11 -o snakeGame snakeGame.cpp -lncursesw**


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

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.

- 1) 제한 요소 : Map은 Window 위에 표현하고 Snake와 Item들까지 함께 화면위에 출력하는 부분
해결 방법 : Map, Snake, Item을 모두 하나의 배열(table)에 넣어 이를 해결하였습니다.
- 2) 제한 요소 : 벽에만 Gate를 생성해주어야 하는 부분
해결 방법 : 벽이 있는 위치들의 좌표를 임의의 배열(wall1)에 저장하고, 배열에 저장되어 있는 좌표에만 Gate가 생성되도록 하여 이를 해결하였습니다.
- 3) 제한 요소 : 정해진 Mission을 모두 통과하였을 때 다음 Map으로 넘어가게 하는 부분
해결 방법 : Mission을 성공하면 각 Mission마다 미리 선언해둔 com 변수가 'v'로 초기화되도록 하고 모든 Mission에 해당하는 com 변수들이 'v'가 되었을 때 changeMap 함수를 호출하는 방식으로 Map 배열을 바꿔주도록 하며 이를 해결하였습니다.
- 4) 제한 요소 : Snake가 Gate를 통과하여 다음 Gate로 나올 때의 방향 설정을 해주는 부분
해결 방법 : Snake가 나와야 하는 Gate의 출구의 상하좌우에 벽이 있는지를 검사하도록 해주고, 이동하던 방향에 벽이 있으면 시계방향을 기준으로 다음 방향을 검사하여 벽이 없는 방향으로 이동하도록 하여 이를 해결하였습니다.
- 5) 제한 요소 : Gate에서 Snake의 몸통이 모두 빠져 나온 뒤에 Gate의 위치를 변경해주는 것
해결 방법 : 게이트에서 빠져나올 때 스네이크의 머리가 게이트로부터 자신의 몸 길이만큼 떨어진 곳에 닿으면, 즉 스네이크가 게이트에서 완전히 빠져나오면 게이트를 다른 곳으로 옮겨주었습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

2.2.5 결과물 목록

작성요령 (5점)

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

- map.h

2차원 배열로 Snake Map의 정보를 담아놓은 헤더 파일이다. Mission이 클리어됨에 따라 난이도가 올라가며 바뀔 Map이 총 5단계로 구성되어 있다. Map 전체를 감싸고 있는 부분은 3, Snake가 접촉할 수 없는 모서리 부분은 2, 일반 벽은 1, Snake가 이동할 수 있는 부분은 1로 표현되어있습니다.

- sclass.h

Snake Game의 전반적인 기능을 구현하기 위해 필요한 것들을 Snake class 안에 다양한 함수들로 구현해 놓은 헤더 파일입니다.

- vector.h


Snake를 표현하고 움직일 때에 사용할 Vector 라이브러리에 연산자 오버로딩을 해주기 위한 헤더 파일이다. Snake와 Item 등등을 구현하며 좌표값을 사용해야 할 때 좌표값끼리 연산을 실행해주기 위하여 연산자 오버로딩을 해 준 파일입니다.

- snake3.cpp

Ncurses 라이브러리를 이용해 Snake Game이 구현된 main c++ 파일이다. sclass.h에서 구현한 다양한 함수들을 이용해 실제로 Snake Game을 구현하는 파일입니다.

- snake3

Snake Game을 플레이하기 위해 snake3.cpp를 컴파일한 make 파일입니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

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.

김하연 :


처음 제가 맡은 것은 스네이크가 윈도우에서 움직이도록 하는 것이었습니다. 조원들이 만들어준 맵을 먼저 화면에 출력하고 그 위에 스네이크를 올리려고 했으나 처음엔 wrefresh를 매번 해주어 딜레이가 생기고 잘 작동하지 않았습니다. 많은 디버깅 끝에 배열 하나에 맵과 스네이크를 넣어 화면에 스네이크와 맵을 오류 없이 출력하게 되었습니다. 그 뒤 스네이크를 키보드 입력에 따라 움직이게 해주었는데 이는 벡터를 이용하였습니다. 어려움이 많았지만 조원들이 많은 도움을 주었습니다. 스네이크를 움직이는데 성공했고 이는 프로젝트를 시작한지 일주일 안에 이루어졌습니다. 그만큼 모두 함께 열심히 했던 것 같습니다. 희망을 본 저희조는 스네이크 프로젝트에 더욱 열의를 가지고 임했습니다. 그 이후로도 매주 조원들과 회의를 했는데 그 때마다 저희는 역할을 분담하였습니다. 저는 그 역할을 최대한 빠르게 수행하고자 노력했습니다. 조원 모두 함께 열심히 하여 저희조는 5월 안에 대부분의 기능을 구현하는데에 성공했습니다. 저는 스네이크 움직이기 이후에도 Apple, Poison, Gate를 구현하는데 일조하였습니다. 조원 모두 코드를 주고 받으며 매일 소통하였고 조원들은 제가 역할을 수행하면 격려와 칭찬을 아끼지 않았습니다. 가장 기억에 남는 부분은 Gate입니다. 가장 오래 걸렸던 부분이기도 합니다. 그래서인지 Gate 구현을 마쳤을 때 가장 뿌듯했습니다. 저는 이번 프로젝트를 하면서 많은 성장을 했다고 생각합니다. c++로 코딩을 하는것에 자신감이 생겼고 제 생각을 코드로 옮기는 것을 정말 못했는데 이번에 많이 나아진 것 같습니다. 조원 모두 열심히 해서 좋은 결과를 얻은 것 같습니다. 조원들에게 감사하고 이런 기회를 제공해 주신 교수님께 감사드립니다..

최혁태 :

저는 주로 map을 구현하고, gate 구현을 도왔습니다. 시작이 반이라고 1단계 맵을 구현하자마자 순식간에 하나 하나 완성되는 모습을 볼 수 있었습니다. 하지만 4단계에 도착했을 때 난관에 봉착했고 이유도 모르는 에러가 막 생겨났습니다. 그때부터는 하루에 한번 회의했다고 해도 과언이 아닐정도로 조원들 모두가 관심을 갖고 문제를 해결하려고 노력했습니다. 에러의 원인을 찾고 고쳐나가는 과정에서 공공 앓던 문제를 하나 해결할때마다 큰 희열을 느꼈고, 다음 단계도 해보고 싶다는 의지의 원동력이 되었던 것 같습니다. 프로젝트를 하면서 수업시간에 다루지 않았던 내용을 스스로 공부하여 구현한다는게 가장 도움이 되었다고 생각합니다. 아무래도 수업시간에는 ncurses라던지, 사소한것 하나하나 다 배우고 넘어가는게 사실상 불가능하지만, 그 부분을 프로젝트로 잘 메운 것 같습니다.

윤민상 :

가장 처음 제가 맡았던 역할은 구현된 Map과 Snake를 합쳐 원하는 Map 위에서 Snake Game을 진행시킬 수 있도록 하는 역할이었습니다. 처음 보는 ncurses라는 라이브러리를 사용해야 하기 때문에 초반에는 많이 막막했지만, 모르는 부분을 팀원들에게 배워가며 차근차근 프로젝트를 같이 진행해 나가다 보니 점차 Ncurses에 익숙해 지며 제 자신이 많은 발전을 할 수 있었던 것 같습니다. 그 이후로

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

부가적인 기능들을 추가하는 역할을 진행해 가면서도, 모르는 부분도 많았지만 뛰어난 팀원들에게 많은 것을 배우며 프로젝트를 진행했기 때문에 원하는 기능을 구현할 수 있었던 것 같아 팀원들에게도 감사한 마음이 매우 큼니다. 수업시간에 배우고 실습을 해 보았던 예제를 다양한 코드들에 적용하여 원하는 기능을 구현한다는 것 또한 상당히 어려웠지만 전반적으로 코딩 실력에 많은 도움이 되었던 것 같습니다. 이번 프로젝트에서는 Ncurses라는 새로운 라이브러리도 많이 다뤄보았고, C++로 처음 참여해본 프로젝트였기 때문에 많은 것을 얻어가는 굉장히 의미 있는 시간이었던 것 같습니다.

4 참고 문헌


번호	종류	제목	출처	발행 년도	저자	기타
1	웹페이지	vector - C++ Reference - Cplusplus.com	http://www.cplusplus.com/reference/vector/vector/	X	X	공식 c++ vector 레퍼런스 사이트
2	웹페이지	[C++]vector container 정리 및 사용법	https://blockdmask.tistory.com/70	2017 년도	Block DMask	
3	웹페이지	ncurses 프로그래밍	https://psman2.tistory.com/entry/ncurses-%ED%94%84%E%B%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D	2011 년도	윤상배	
4	웹페이지	[cpp] operator overloading (연산자 오버로딩)	http://blog.naver.com/PostView.nhn?blogId=kyed203&logNo=220151651363	2014 년도	memOry	

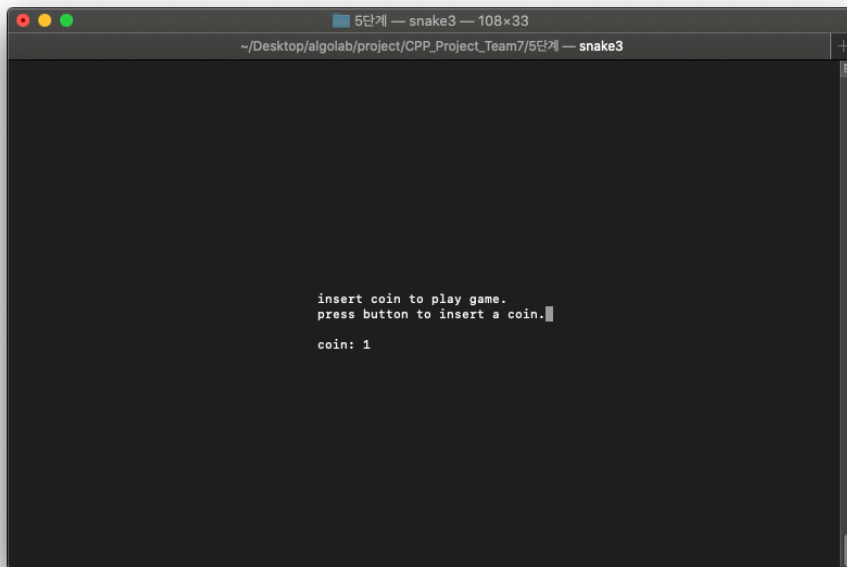
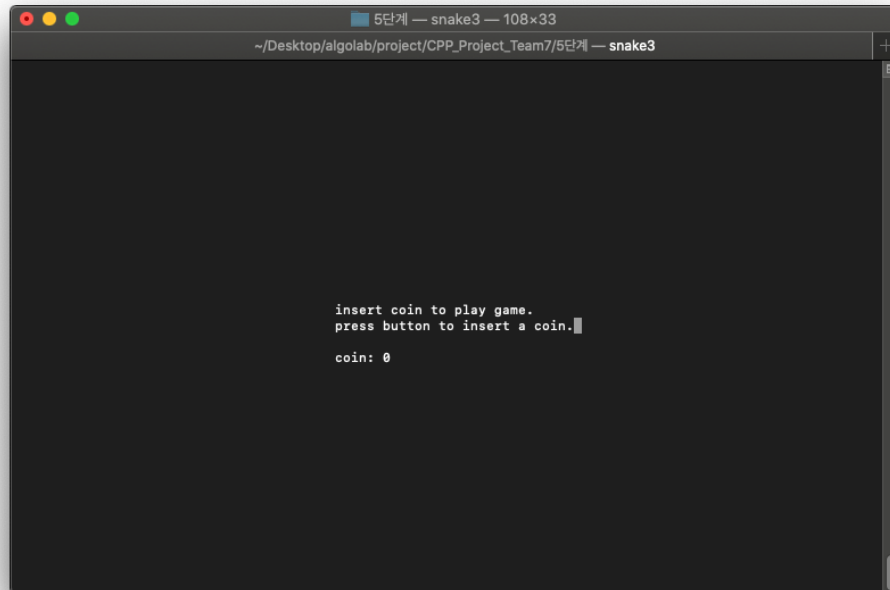
5 부록

작성요령 (15점)

프로젝트의 결과물을 사용하기 위한 방법에 대해서 작성하세요.

5.1 사용자 매뉴얼

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22

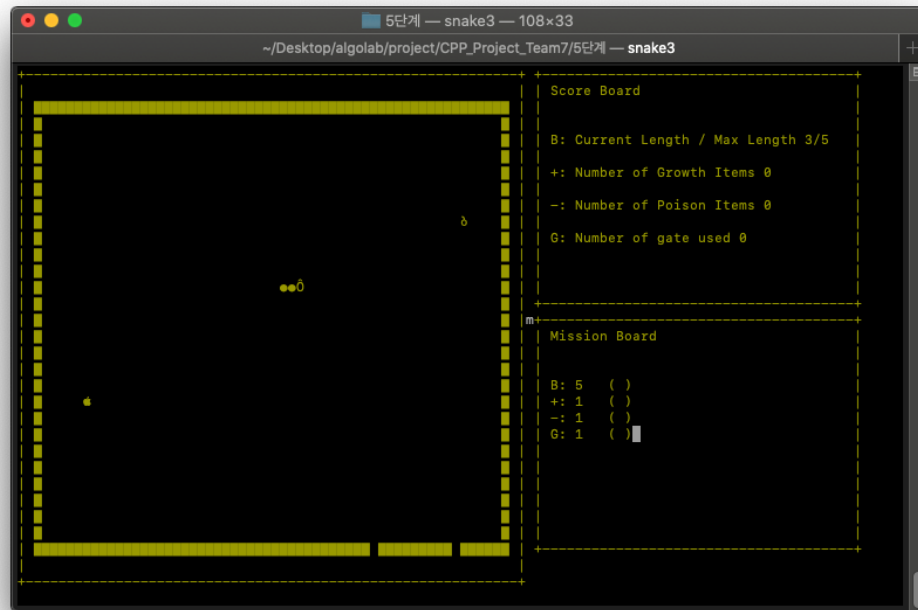


처음 게임을 시작했을 때 나오는 화면입니다. 이 화면에서 키보드 enter 버튼을 입력받을 시 coin의 개수가 1 늘어나며 게임이 시작됩니다.


결과보고서	
프로젝트 명	Snake Game
팀 명	7조
Confidential Restricted	Version 1.2
	2020x-JUN-22

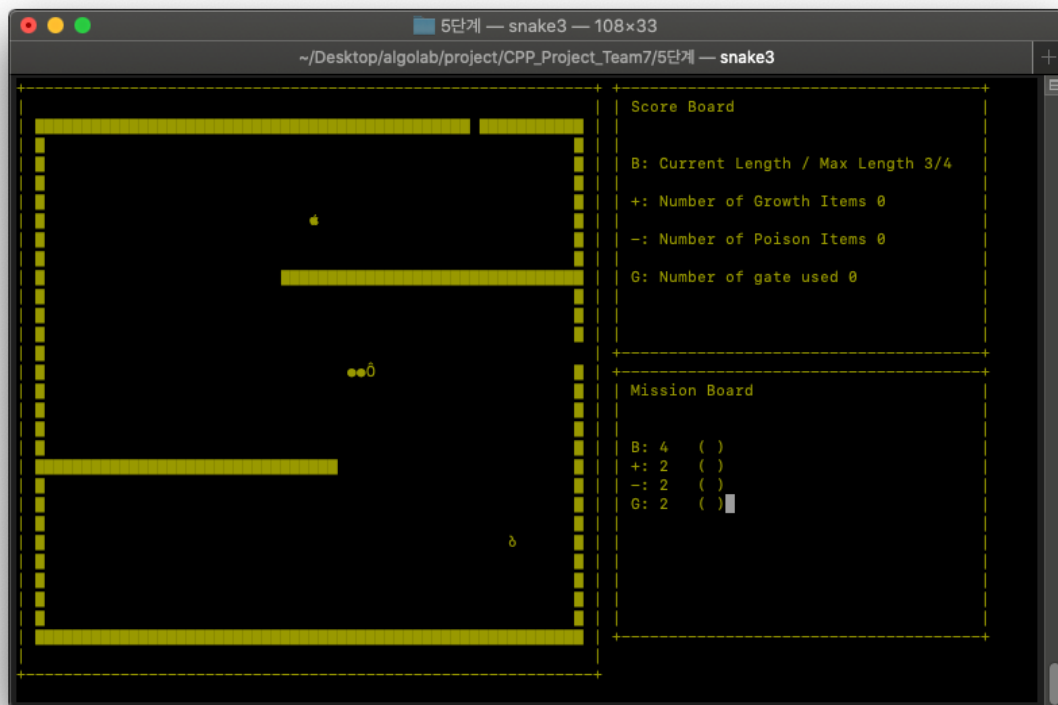


국민대학교
컴퓨터공학부
캡스톤 디자인 I




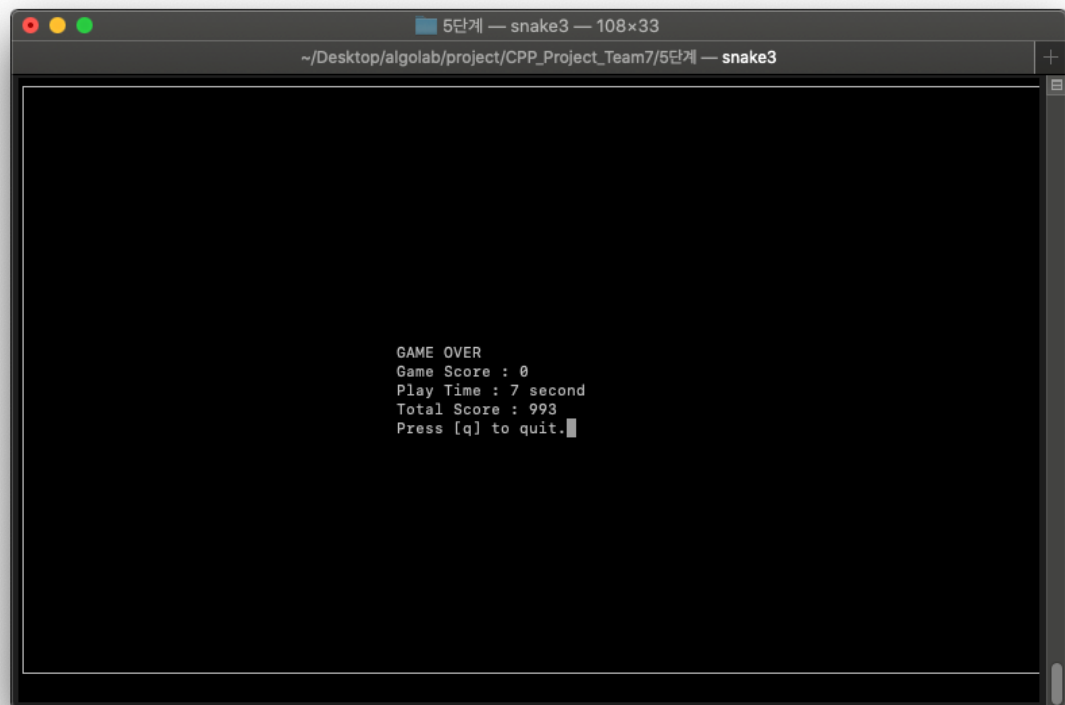
게임이 시작된 화면입니다. 방향키를 통해 Snake의 이동 방향을 자유자재로 바꿀 수 있으며, 키보드의 'p' 버튼을 누를시 게임을 일시중지 시킬 수 있고, 'q' 버튼을 누를시 게임을 종료할 수 있습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22




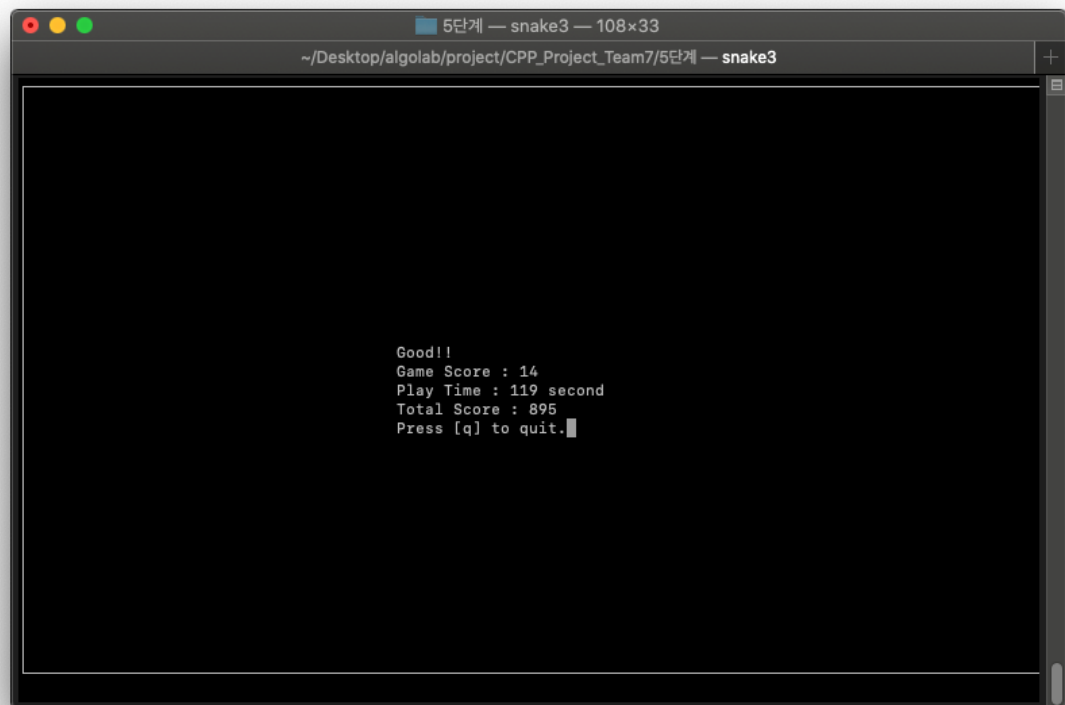
미션을 clear 할때마다 단계가 올라가며 맵이 바뀌는걸 볼 수 있습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22



Snake가 벽에 부딪혔거나, 머리의 진행 방향의 반대 방향을 입력받는 등 다양한 이유로 게임 오버가 되었을 때 나오는 화면입니다. 키보드의 enter 키를 입력받을 시 게임을 다시 시작할 수 있고, 'q' 버튼을 누를 시 게임을 종료할 수 있습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Snake Game	
	팀 명	7조	
	Confidential Restricted	Version 1.2	2020x-JUN-22



게임을 클리어 했을 시 나오는 화면입니다. 훌륭한 실력으로 게임을 클리어 하였다는 증거입니다!

성공했다는 문구(Good!!)와 게임을 끝내는데 걸린 총 시간, 총 점수 등을 보여줍니다.

5.2 설치 방법

main c++ 파일인 snake3.cpp를 컴파일을 해주어야 합니다.

g++ -o make파일명 snake3.cpp -lncurses 의 명령어로 컴파일 해준 후, 생성된 make 파일을 실행해주어야 합니다.