

과 목 : 2019 전공기초프로젝트1

화 · 목 반 1조

담당교수 : 차 리 서

제출기한 : 2019. 04. 28

2019

전공기초프로젝트1

1차 검사 보고서

- 미로 찾기 프로그램 -

소프트웨어학과 1조

김동진 201714150

박민기 201714151

박종현 201714152

허승희 201714158

목 차

1. 메인메뉴
 - 1-1. 번호입력
2. 게임시작
 - 2-1. 맵 선택
 - 2-2. 텔레포트
3. 맵 세팅
 - 3-1. 번호입력
 - 3-2. 맵 확인
 - 맵 선택
 - 상위메뉴
 - 3-3. 맵 추가
 - 맵 선택
4. 단위테스트

설 명

사용자가 프로그램과 interaction하는 부분에 대한 test case들을 작성하였으며 예외처리 실패 부분과 설계가 어긋난 부분을 빨간 색깔로 구분을 하였다.

1. 메인메뉴

- 메인 메뉴에서 이동하려는 하위메뉴 및 기능에 대한 번호를 입력 시 1,2,3이 아닌 숫자, 영어, 한글, 특수기호, 공백이 있다면 에러메시지를 표시한 후 다시 한 번 입력할 것을 요구한다.

- 각 testcase를 입력시에 화면 초기화되어 testcase별로 결과값을 캡처하였다.

testcase1 : 1,2,3이 아닌 숫자

return : 에러메시지 표출

```
*****
1 - Game Start.
2 - Map Setting.
3 - Exit.
*****
선택하세요:55
잘못된 입력
다시입력하세요
```

testcase2 : 한글

return : 에러메시지 표출

```
*****
1 - Game Start.
2 - Map Setting.
3 - Exit.
*****
선택하세요:ㅇ
잘못된 입력
다시입력하세요
```

testcase3 : 공백

return : 에러메시지 표출

```
*****
1 - Game Start.
2 - Map Setting.
3 - Exit.
*****
선택하세요:
잘못된 입력
다시입력하세요
```

testcase4 : 특수기호

return : 에러메시지 표출

```
*****
1 - Game Start.
2 - Map Setting.
3 - Exit.
*****
선택하세요:#
잘못된 입력
다시입력하세요
```

testcase5 : 영어

return : 에러메시지 표출

```
*****
1 - Game Start.
2 - Map Setting.
3 - Exit.
*****
선택하세요:A
잘못된 입력
다시입력하세요
```

testcase6 : 1

return : 지도 입력 요구

```
*****
1 - Game Start.
2 - Map Setting.
3 - Exit.
*****
선택하세요:1
지도를 입력하세요 :
```

testcase7 : 2

return : 화면 초기화 후 하위메뉴 이동

```
*****
1 - Game Start.
2 - Map Setting.
3 - Exit.
*****
선택하세요:2
```

```
*****
1 - 맵 확인.
2 - 맵 추가.
3 - 상위 메뉴.
*****
선택하세요:
```

testcase8 : 3

return : 종료

```
*****
1 - Game Start.
2 - Map Setting.
3 - Exit.
*****
선택하세요:3
게임 종료.
계속하려면 아무 키나 누르십시오 . . .
```

2. 게임시작

2.1 맵선택

- 올바른 형식의 입력(ex: 홍길동.txt)이 아닐 경우 에러메시지를 표시한다.
올바른 형식으로 입력을 하였으나 지도가 디렉토리안에 없는 경우에도
러메시지를 호출하며 둘의 에러메세지는 다른 형태의 메시지로 출력한다.

· 잘못된 입력(.txt 형식이 아닌 경우)

testcase1 : 숫자

return : 에러메시지 표출

testcase2 : 영어

return : 에러메시지 표출

testcase3 : 공백

return : 에러메시지 표출

testcase4 : 특수문자

return : 에러메시지 표출

testcase5 : 한글

return : 에러메시지 표출

testcase6 : 숫자+영어+한글+특수문자

return : 에러메시지 표출

```

*****
1 - Game Start.
2 - Map Setting.
3 - Exit.
*****
선택하세요:1
지도를 입력하세요 : 1
잘못된 입력하세요 : j
지도를 입력하세요 :
잘못된 입력하세요 : @
지도를 입력하세요 : 1
잘못된 입력하세요 : 1d1.tx
잘못된

```

· 파일이 없는 경우

testcase1 : 올바른 형식(ex: 홍길동)의 파일 형식 입력 후 디렉토리에 없는 파일을 입력

return : 화면 초기화 후 에러 메시지 출력

```

*****
1 - Game Start.
2 - Map Setting.
3 - Exit.
*****
선택하세요:1
지도를 입력하세요 : 홍길동.txt

```

지도를 찾을 수 없습니다!

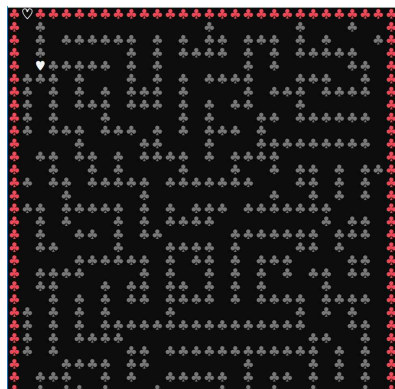
지도를 입력하세요 :

· 올바른 형식

testcase : 1.txt

return : 맵 출력 및 게임시작

지도를 입력하세요 : 1.txt



2.2 텔레포트

- 갈림길을 만나고 이동회수가 15회수 이상일 때, 사용자에게 텔레포트 여부를 묻는다. 사용한다 하면 지나온 좌표 한에 그 좌표로 텔레포트를 할 수 있다.

• **텔레포트 여부** : y/n이 아닌 영어, 숫자, 공백, 특수문자, 한글을 입력 시 에러메시지 표출

testcase1 : 숫자

return : 에러메시지 표출

testcase2 : 영어

return : 에러메시지 표출

testcase3 : 공백 (예외처리 실패)

return : 무반응

testcase4 : 특수문자

return : 에러메시지 표출

testcase5 : 한글

return : 에러메시지 표출

testcase6 : 숫자+영어+한글+특수문자

return : 에러메시지 표출



testcase7 : y

return : 좌표 입력 요구

```
텔레포트 사용하실? (Y/N)
n
텔레포트 사용하실? (Y/N)
y
좌표 입력해 ....
```

testcase8 : n

return : 길찾기 재개

```
텔레포트 사용하실? (Y/N)
n
텔레포트 사용하실? (Y/N)
```

• **텔레포트 사용** : 갈 수 없는 좌표와 영어, 공백, 특수문자, 한글을 입력 시
에러메시지 표출

testcase1 : 갈 수 없는 좌표

return : 에러메시지 표출

```
텔레포트 사용하실? (Y/N)
y
좌표 입력해 ....
400 20

갈 수 없는 곳이야
좌표 입력해 ....
```

testcase2 : 영어

return : 에러메시지 표출

testcase3 : 공백 (예외처리 실패)

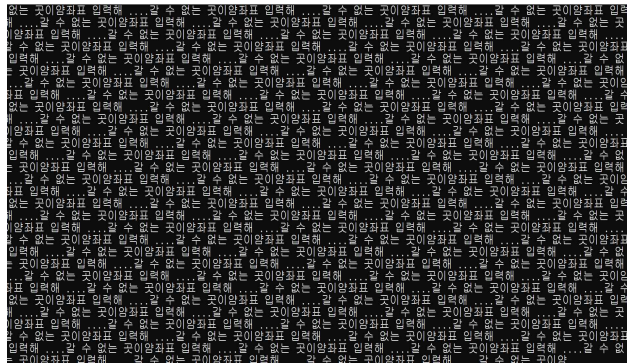
return : 무반응

testcase4 : 특수문자

return : 에러메시지 표출

testcase5 : 한글

return : 에러메시지 표출



3. 메인메뉴

3.1 번호입력

- 메뉴에서 이동하려는 기능에 대한 번호를 입력 시 1,2,3이 아닌 숫자, 영어, 한글, 특수기호, 공백이 있다면 에러메시지를 표시한 후 다시 한 번 입력할 것을 요구한다.

- 각 testcase를 입력시에 화면 초기화되어 testcase별로 결과값을 캡쳐하였다.

testcase1 : 1,2,3이 아닌 숫자

return : 에러메시지 표출

```
*****
1 - 맵 확인.
2 - 맵 추가.
3 - 상위 메뉴.
*****
선택하세요:55
잘못된 입력
다시입력하세요
```

testcase2 : 한글

return : 에러메시지 표출

```
*****
1 - 맵 확인.
2 - 맵 추가.
3 - 상위 메뉴.
*****
선택하세요:ㅇㅅㅇ
잘못된 입력
다시입력하세요
```

testcase3 : 공백

return : 에러메시지 표출

```
*****
1 - 맵 확인.
2 - 맵 추가.
3 - 상위 메뉴.
*****
선택하세요:
잘못된 입력
다시입력하세요
```

testcase4 : 특수기호

return : 에러메시지 표출

```
*****
1 - 맵 확인.
2 - 맵 추가.
3 - 상위 메뉴.
*****
선택하세요: @@@
잘못된 입력
다시입력하세요
```

testcase5 : 영어

return : 에러메시지 표출

```
*****
1 - 맵 확인.
2 - 맵 추가.
3 - 상위 메뉴.
*****
선택하세요: idontwantodothis
잘못된 입력
다시입력하세요
```

testcase6 : 1

return : 지도 입력 요구

```
*****
1 - 맵 확인.
2 - 맵 추가.
3 - 상위 메뉴.
*****
선택하세요: 1
지도를 입력하세요 :
```

testcase7 : 2

return : 화면 초기화 맵 추가 절차 화면 출력

```
*****
1 - 맵 확인.
2 - 맵 추가.
3 - 상위 메뉴.
*****
선택하세요:2
```

맵을 추가 하시겠습니까? (-1을 입력시 상위메뉴, 아무 키 입력시 맵 추가 절차 시작)

testcase8 : 3

return : 화면 초기화 후 상위 메뉴 화면 출력

```
*****
1 - 맵 확인.
2 - 맵 추가.
3 - 상위 메뉴.
*****
선택하세요:3
```

```
*****
1 - Game Start.
2 - Map Setting.
3 - Exit.
*****
선택하세요:
```

3.2 맵 확인

- 맵 선택 :

올바른 형식의 입력(ex: 홍길동.txt)이 아닐 경우 에러메시지를 표시한다.
올바른 형식으로 입력을 하였으나 지도가 디렉토리안에 없는 경우에도
러메시지를 호출하며 둘의 에러메세지는 다른 형태의 메시지로 출력한다.

· 잘못된 입력(.txt 형식이 아닌 경우)

testcase1 : 숫자

return : 에러메시지 표출

testcase2 : 영어

return : 에러메시지 표출

testcase3 : 공백

return : 에러메시지 표출

testcase4 : 특수문자

return : 에러메시지 표출

testcase5 : 한글

return : 에러메시지 표출

testcase6 : 숫자+영어+한글+특수문자

return : 에러메시지 표출

```
*****
1 - 맵 확인.
2 - 맵 추가.
3 - 상위 메뉴.
*****
선택하세요:1
지도를 입력하세요 : 555
잘못된 입력
지도를 입력하세요 : anyoung
잘못된 입력
지도를 입력하세요 :
잘못된 입력
지도를 입력하세요 : (
잘못된 입력
지도를 입력하세요 : 하
잘못된 입력
지도를 입력하세요 : 2jㅇ.tx
잘못된 입력
지도를 입력하세요 :
```

· 파일이 없는 경우

testcase1 : 올바른 형식(ex: 홍길동)의 파일 형식 입력 후 디렉토리에 없는 파일을 입력

return : 화면 초기화 후 에러 메시지 출력

지도를 입력하세요 : sl.txt

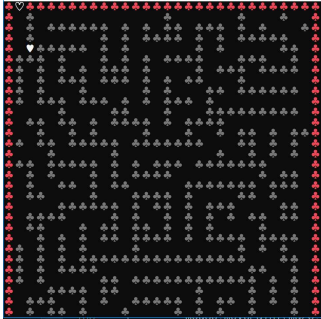
지도를 찾을 수 없습니다!

지도를 입력하세요 :

· 올바른 형식

testcase : 1.txt

return : 맵 출력



– 상위 메뉴가기 :

올바른 형식의 입력인 -1을 받을 때 까지 현재 상태에 머물고 -1을 받으면 올바른 선택으로 간주하여 현재 화면 초기화 후 상위 메뉴로 이동한다.

· 잘못된 입력(-1이 아닌 경우)

testcase1 : 숫자

return : 현재 화면에 머물

testcase2 : 영어

return : 현재 화면에 머물

testcase3 : 공백

return : 현재 화면에 머물

testcase4 : 특수문자

return : 현재 화면에 머물

testcase5 : 한글

return : 현재 화면에 머물

testcase6 : 숫자+영어+한글+특수문자

return : 현재 화면에 머물



• 올바른 입력

testcase1 : -1

return : 현재 화면 초기화 후 상위 메뉴로 이동



3.3 맵 추가

- 사용자에게 미로를 입력 받는다. (0(길), 1(벽), 2(입구), 3(출구))

다음의 경우에 대하여 예외처리를 한다.

길이 하나도 없는 경우, 벽이 하나도 없는 경우, 입구가 하나가 아닌 경우,

출구가0개인경우, 출구가 5개보다 많은 경우,

맨위에벽이거나입구가아니거나출구가아닌길이있는경우,

맨왼쪽에벽이거나입구가아니거나출구가아닌길이있는경우,

맨오른쪽에벽이거나입구가아니거나출구가아닌길이있는경우,

맨밑에벽이거나입구가아니거나출구가아닌길이있는경우

0,1,2,3이아닌다른값을받는 경우.

파일이름 입력받고 파일을 저장한다.

· 행/열 입력

testcase1 : 숫자

return : 현재 화면에 머뭄

testcase2 : 영어

return : 현재 화면에 머뭄

testcase3 : 공백

return : 공백 인식 실패

testcase4 : 특수문자

return : 현재 화면에 머뭄

testcase5 : 한글

return : 현재 화면에 머뭄

testcase6 : 숫자+영어+한글+특수문자

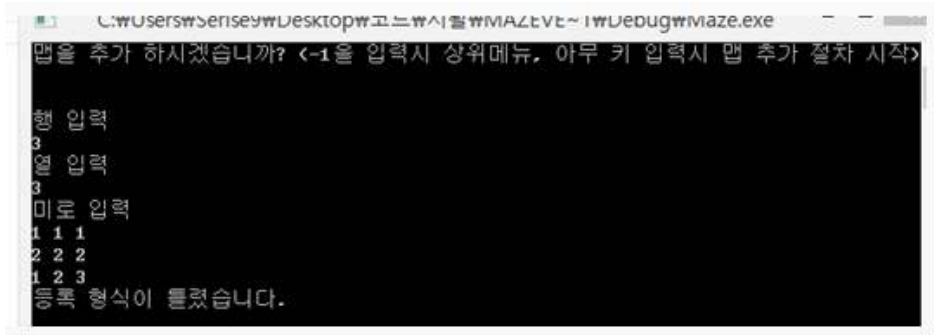
return : 미로 입력 요구 (예외처리 실패)

```
맵을 추가 하시겠습니까? (-1을 입력시 상위메뉴, 아무 키 입력시 맵 추가 절차 시작)
행 입력
999
열 입력
999
0에서 50 사이의 행을 입력하세요
맵을 추가 하시겠습니까? (-1을 입력시 상위메뉴, 아무 키 입력시 맵 추가 절차 시작)
행 입력
k
열 입력
k
0에서 50 사이의 행을 입력하세요
맵을 추가 하시겠습니까? (-1을 입력시 상위메뉴, 아무 키 입력시 맵 추가 절차 시작)
(행) 입력
s
(열) 입력
d
0에서 50 사이의 행을 입력하세요
맵을 추가 하시겠습니까? (-1을 입력시 상위메뉴, 아무 키 입력시 맵 추가 절차 시작)
#
(행) 입력
#
0에서 50 사이의 행을 입력하세요
맵을 추가 하시겠습니까? (-1을 입력시 상위메뉴, 아무 키 입력시 맵 추가 절차 시작)
(행) 입력
0
(열) 입력
0
0에서 50 사이의 행을 입력하세요
맵을 추가 하시겠습니까? (-1을 입력시 상위메뉴, 아무 키 입력시 맵 추가 절차 시작)
(행) 입력
3k
(열) 입력
t#
(행) 입력
31
(열) 입력
z@
미로 입력
```

· 미로 입력(3x3 행열)

testcase1 : 길(0)이 하나도 없는 경우

return : 에러 메시지 표시

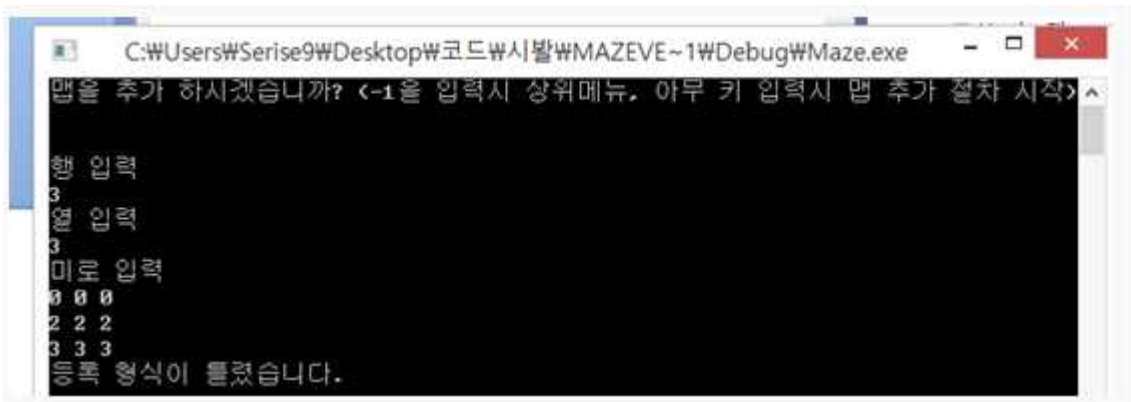


```
C:\Users\serise9\Desktop\코드\시뮬\MAZEVE~1\Debug\Maze.exe
맵을 추가 하시겠습니까? <-1을 입력시 상위메뉴, 아무 키 입력시 맵 추가 절차 시작>

행 입력
3
열 입력
3
미로 입력
1 1 1
2 2 2
1 2 3
등록 형식이 틀렸습니다.
```

testcase2 : 벽(1)이 하나도 없는 경우, 맨 위에 벽(1)이거나 입구(2)가 아니거나 출구(3)가 아닌 길(0)이 있는 경우

return : 에러 메시지 표시

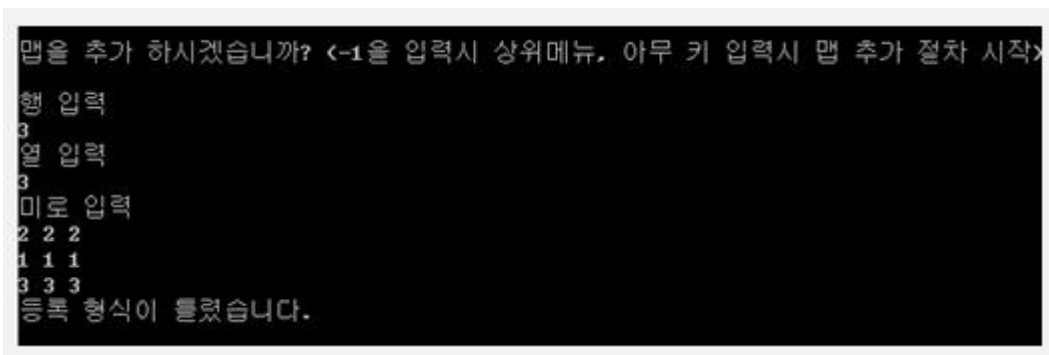


```
C:\Users\serise9\Desktop\코드\시뮬\MAZEVE~1\Debug\Maze.exe
맵을 추가 하시겠습니까? <-1을 입력시 상위메뉴, 아무 키 입력시 맵 추가 절차 시작>

행 입력
3
열 입력
3
미로 입력
0 0 0
2 2 2
3 3 3
등록 형식이 틀렸습니다.
```

testcase3 : 입구(2)가 하나가 아닌 경우

return : 에러 메시지 표시



```
C:\Users\serise9\Desktop\코드\시뮬\MAZEVE~1\Debug\Maze.exe
맵을 추가 하시겠습니까? <-1을 입력시 상위메뉴, 아무 키 입력시 맵 추가 절차 시작>

행 입력
3
열 입력
3
미로 입력
2 2 2
1 1 1
3 3 3
등록 형식이 틀렸습니다.
```

testcase4 : 출구(3)이 하나도 없는 경우

return : 에러 메시지 표시


```

행 입력
3
열 입력
3
미로 입력
1 2 1
1 0 1
1 0 1
등록 형식이 틀렸습니다.

```

testcase5 : 출구(3)가 5개보다 많은 경우

return : 에러 메시지 표시

```

맵을 추가 하시겠습니까? <-1을 입력시 상위메뉴, 아무 키 입력시 맵 추가 절차 시작>
행 입력
3
열 입력
3
미로 입력
1 2 1
0 3 3
0 3 3
등록 형식이 틀렸습니다.

```

testcase6 : 맨 왼쪽에 벽(1)이거나 입구(2)가 아니거나 출구(3)가 아닌 길(0)이 있는 경우

return : 에러 메시지 표시

```

맵을 추가 하시겠습니까? <-1을 입력시 상위메뉴, 아무 키 입력시 맵 추가 절차 시작>
행 입력
3
열 입력
3
미로 입력
0 2 1
0 0 1
0 3 1
등록 형식이 틀렸습니다.

```

testcase7 : 맨 오른쪽에 벽(1)이거나 입구(2)가 아니거나 출구(3)가 아닌 길(0)이 있는 경우

return : 에러 메시지 표시

```

행 입력
3
열 입력
3
미로 입력
1 2 0
1 0 0
1 3 0
등록 형식이 틀렸습니다.

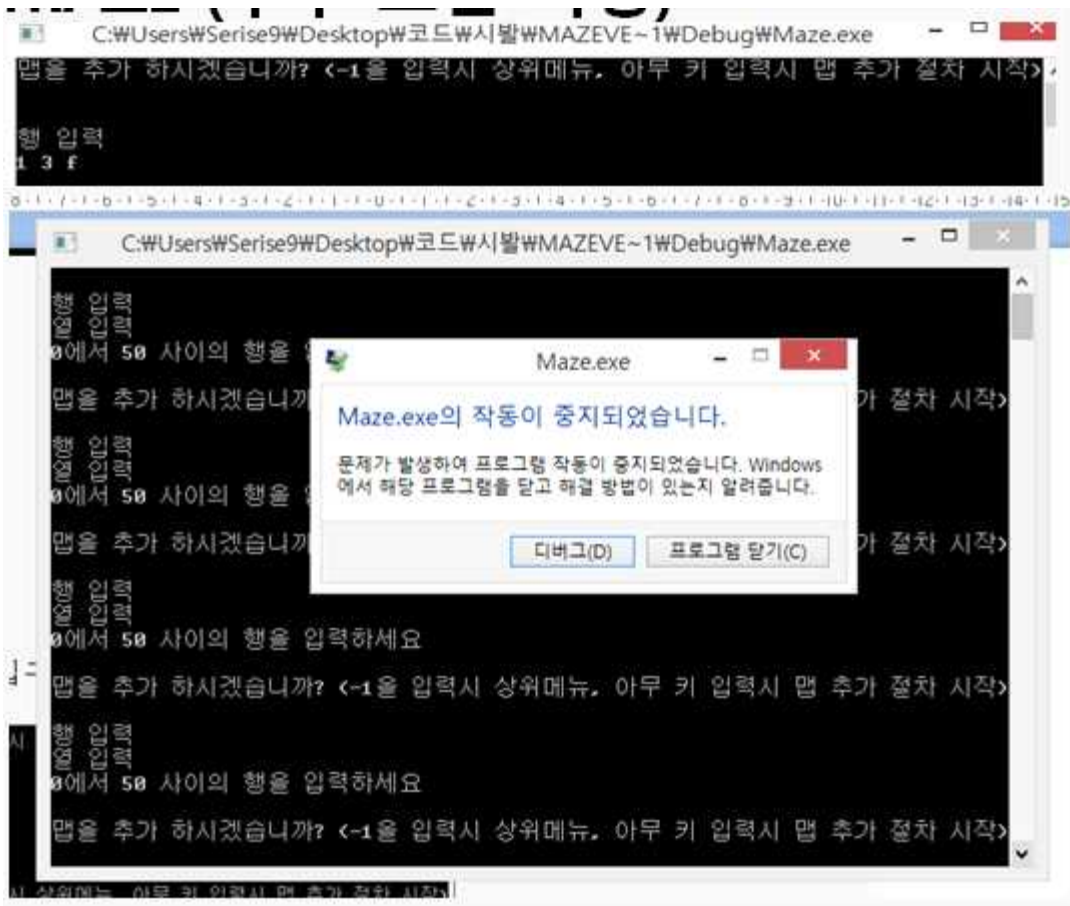
```

testcase8 : 맨 밑에 벽(1)이거나 입구(2)가 아니거나 출구(3)가 아닌 길(0)이 있는 경우

return : 에러 메시지 표시

```
행 입력
3
열 입력
3
미로 입력
1 2 0
1 0 0
1 3 0
등록 형식이 틀렸습니다.
```

testcase9 : 0,1,2,3외 다른 값을 받는 경우
return : 오류 (예외처리 실패)



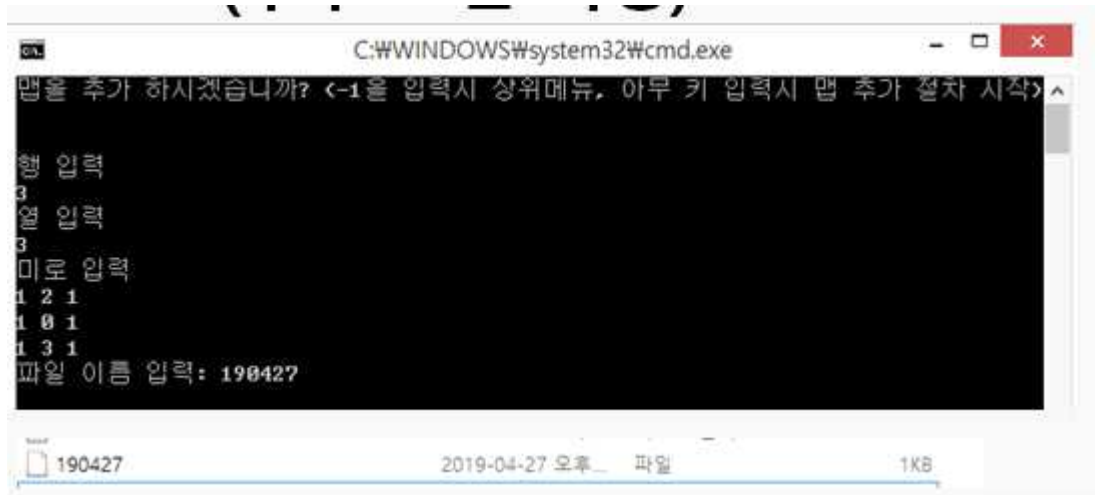
testcase10 : 올바른 미로 형식일 경우
return : 지도이름 입력 요구

```
맵을 추가 하시겠습니까? (-1을 입력시 상위메뉴, 아무 키 입력시 맵 추가 절차 시작)
3
행 입력
3
열 입력
3
미로 입력
1 2 1
1 0 1
1 3 1
파일 이름 입력:
```

- 파일이름 입력

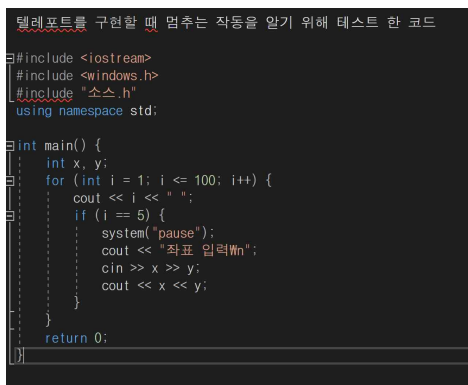
testcase : 올바른 파일 형식 아닌 값 입력(.txt파일형식이 아닌 경우
예외처리 필요)

return : 지도 생성

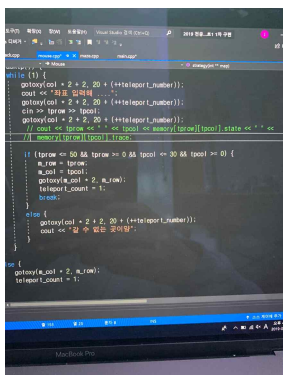


4. 단위테스트

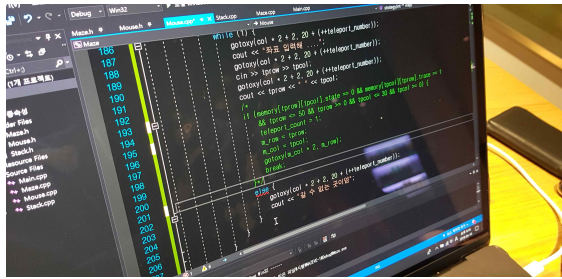
- 텔레포트 구현할 때 멈추는 작동 알기 위한 테스트



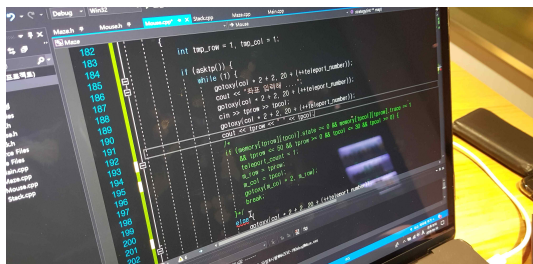
- 텔레포트 후 위치 파악을 위한 단위테스트



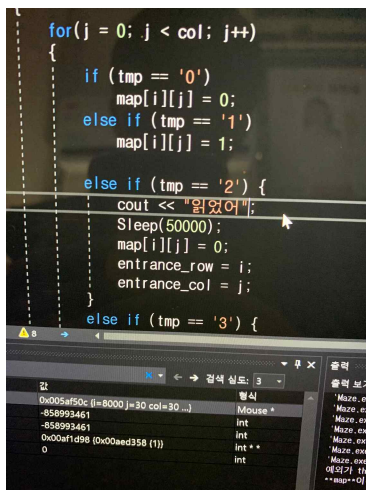
- 텔레포트 구현할 때 유효한 값 체크 위한 테스트



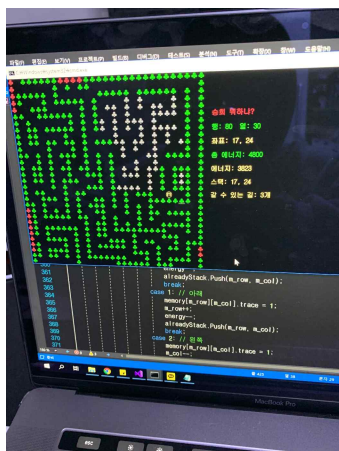
- 텔레포트 구현할 때 갈림길 테스트



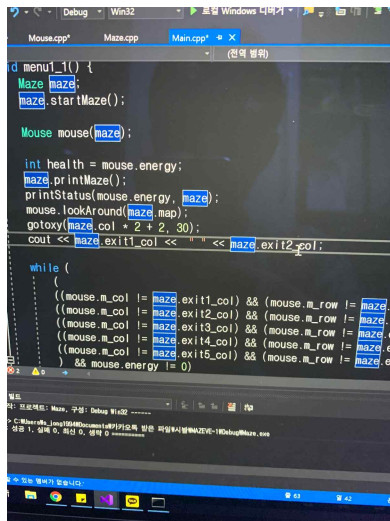
- 맵에서 입구(2)를 읽을 때 테스트 하였던 코드



- 스택이 제대로 작동하는지 테스트



• 게임종료 조건 작동 테스트



• 메뉴 입력값 중 1,2,3이 아닌 default 값들 확인하는 테스트

```
default: {
    cout << "잘못된 입력\n다시입력하세요\n";
    sleep(1000);
    system("cls");
    menu1();
}
```

• makeMaze()에대한 테스트 하였던 코드 및 주석

```
/*
makeMaze()

makeMaze() 미로만들기함수사용자에게행과열을받는다.
사용자에게미로를입력받는다. ( 0(길), 1(벽), 2(입구), 3(출구) )
다음의경우에대하여예외처리를한다.
길이하나도없는경우, 벽이하나도없는경우, 입구가하나가아닌경우,
출구가0개인경우, 출구가5개보다많은경우,
맨위에벽이거나입구가아니거나출구가아닌길이있는경우,
맨왼쪽에벽이거나입구가아니거나출구가아닌길이있는경우,
맨오른쪽에벽이거나입구가아니거나출구가아닌길이있는경우,
맨밑에벽이거나입구가아니거나출구가아닌길이있는경우
0,1,2,3이아닌다른값을받는경우
파일이름입력받고파일저장한다.
*/

void Maze::makeMaze() { // 미로만들기
```

```

char mazeName[20]; // maze를만들때이름을저장하는변수

int mat[50][30]; // 미로행50 열30
int c, r;

int zero = 0; // 초기길값0
int one = 0; // 초기벽값0
int two = 0; // 초기입구값0
int three = 0; // 초기출구값0

string cc = " " // 열값을받는string 변수
string rr = " " // 행값을받는string 변수
string get = " " // 사용자의입력값을받는string 변수

bool escape = true // while문탈출을위한bool 변수

cout << "맵을추가하시겠습니까을입력시상위메뉴, 아무키입력시맵추가절차시작) \n"

getline(cin, get);

int num = atoi(get.c_str()); // str을int 값으로바꿈

if (num == -1) goto GOODBYE; // -1 이면goto 문을이용해서상위메뉴로감

else {

cout << "행입력\n" // 행입력
cin >> rr;
r = atoi(rr.c_str());

cout << "열입력\n" // 열입력
cin >> cc;
c = atoi(cc.c_str());

while (1) {

if (r > 0 && r <= 50) break // 행이0에서50 사이이면break
else { // 0에서50 사이입력안하면다시입력한다.
cout << "0에서50 사이의행을입력하세요" << endl;

```

```

cout << endl;
makeMaze();
break
}
}

while (1) {
if (c > 0 && c <= 50) break // 열이0에서30 사이이면break
else { // 0에서30 사이입력안하면다시입력한다.
cout << "0과30사이의열을입력하세요" << endl;
cout << endl;
makeMaze();
break

}

}

while (1) {
cout << "미로입력\n"
for (int i = 0; i < r; i++)
{
for (int j = 0; j < c; j++)
{
cin >> mat[i][j];

if (mat[i][j] == 0) { // 길+1
zero++;
}
else if (mat[i][j] == 1) { // 벽+1
one++;
}
else if (mat[i][j] == 2) { // 입구+1
two++;

}
else if (mat[i][j] == 3) { // 출구+1
three++;

}
}
}
}

```

```

}
}
if (zero == 0) { // 길이하나도없는경우에다시한다.
cout << "등록형식이틀렸습니다." << endl;
cout << endl;
makeMaze();
}
else if (one == 0) { // 벽이하나도없는경우에다시한다.
cout << "등록형식이틀렸습니다." << endl;
cout << endl;
makeMaze();
}
else if (two != 1) { // 입구가하나가아닌경우에다시한다.
cout << "등록형식이틀렸습니다." << endl;
cout << endl;
makeMaze();
}
else if (three == 0) { // 출구가0개인경우에다시한다.
cout << "등록형식이틀렸습니다." << endl;
cout << endl;
makeMaze();
}
else if (three > 5) { // 출구가5개보다많은경우에다시한다.
cout << "등록형식이틀렸습니다." << endl;
cout << endl;
makeMaze();
}
for (int j = 0; j < c; j++)
{
if (mat[0][j] == 0) { // 맨위에벽이거나입구가아니거나출구가아닌길이있는경우에다시한다.
cout << "등록형식이틀렸습니다." << endl;
cout << endl;
makeMaze();
}
}
for (int i = 0; i < r; i++)
{
if (mat[i][0] == 0) { // 맨왼쪽에벽이거나입구가아니거나출구가아닌길이있는경우에다시한

```



```

다.
cout << "등록형식이틀렸습니다." << endl;
cout << endl;
makeMaze();
}
}
for (int i = 0; i < r; i++)
{
if (mat[i][c] == 0) { // 맨오른에벽이거나입구가아니거나출구가아닌길이있는경우에다시한다.
다.
cout << "등록형식이틀렸습니다." << endl;
cout << endl;
makeMaze();
}
}
for (int j = 0; j < c; j++)
{
if (mat[r][j] == 0) { // 맨밑에벽이거나입구가아니거나출구가아닌길이있는경우에다시한다.
cout << "등록형식이틀렸습니다." << endl;
cout << endl;
makeMaze();

}
}
for (int i = 0; i < r; i++)
{
for (int j = 0; j < c; j++)
{
if (mat[i][j] == 1 || mat[i][j] == 2 || mat[i][j] == 0 || mat[i][j] == 3) // 0,1,2,3이아닌
다른값을받는경우에다시한다.
continue
else escape = false
}
}
if (escape == true) break
else escape = true
}

// 파일이름입력받고파일을만든다.

```

```
cout << "파일이름입력: "  
cin >> mazeName;  
  
ofstream outFile(mazeName);  
  
for (int i = 0; i < r; i++)  
{  
    for (int j = 0; j < c; j++)  
    {  
        outFile << mat[i][j];  
    }  
    outFile << "\n"  
}  
cout << "저장되었습니다." << endl;  
}  
  
GOODBYE:  
cout << ""  
  
}
```