Mud_game 보고서

183014 이해건

1. 서론

   A. 프로젝트 목적 및 배경 : c++ 강의 두번째 프로젝트형 실습을 위하여 진행

   B. 목표 : 간단한 mud게임의 구현 및 클라이언트 요구에 따른 develop.

2. 요구사항

   A. 사용자 요구사항 : 유저가 상하좌우로 이동하며 목적지에 도착하면 clear.

   B. 기능 계획

      i. 사용자에게 "상", "하", "좌", "우", "지도", "종료" 중 하나를 입력 받기

         1. 각각의 값 입력 시 해당 방향으로 이동 후 진행상황인 지도를 출력

         2. 지도를 입력 시 전체 지도와 함께 현재 위치와 체력을 출력

         3. 이 중 다른 것을 입력한다면 error 메세지 출력 후 재 입력을 위한 프로세스 진행

      ii. 지도 밖으로 이동하라는 명령을 입력 시 에러 메시지를 출력

      iii. 목적지로 도착하면 성공 메시지를 출력하고 게임을 종료

      iv. 유저의 체력 20이 있고 이동시마다 체력 1 감소, hp가 0 이하가 된다면 실패를 출력하고 게임 종료

   C. 함수 계획

      i. 메인 함수 : while 반복문을 통하여 유저에게 입력을 반복적으로 받으며 그 값에 따라 함수를 호출하면서 게임을 진행

      ii. checkXY : 사용자 캐릭터의 현 위치를 확인하며 적절하면 true, 부적절하면 false를 출력

      iii. displayMap : 전체 지도와 함께 사용자의 현재 위치를 출력하는 함수

      iv. checkGoal : 유저의 현재 위치가 목적지인지를 확인하고, 목적지 값과 동일하다면 true 값을 출력

      v. checkState : 유저의 위치에 있는 아이템을 출력하고 해당 값을 hp에 적용

3. 설계 및 구현

A. 기능 별 구현 사항 :

i. 사용자에게 상하좌우, 지도, 종료 를 입력받기

1. 각각의 값 입력시 해당 방향으로 이동 후 진행상황인 지도를 출력

```
41          if (user_input == "상") {
42              // 위로 한 칸 올라가기
43              user_y -= 1;
44              bool inMap = checkXY(user_x, mapX, user_y, mapY);
45              if (inMap == false) {
46                  cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
47                  user_y += 1;
48              }
49              else {
50                  cout << "위로 한 칸 올라갑니다." << endl;
51                  displayMap(map, user_x, user_y);
52                  user_hp--;
53                  checkState(map, user_x, user_y);
54                  if (user_hp <= 0) {
55                      cout << "HP가 0 이하가 되었습니다. 실패했습니다." << endl;
56                      cout << "게임을 종료합니다." << endl;
57                      break;
58                  }
59                  cout << "현재 HP : " << user_hp <<"  ";
60              }
61          }
```

```
62          else if (user_input == "하") {
63              // TODO: 아래로 한 칸 내려가기
64              user_y += 1;
65              bool inMap = checkXY(user_x, mapX, user_y, mapY);
66              if (inMap == false) {
67                  cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
68                  user_y -= 1;
69              }
70              else {
71                  cout << "위로 한 칸 내려갑니다." << endl;
72                  displayMap(map, user_x, user_y);
73                  user_hp--;
74                  checkState(map, user_x, user_y);
75                  if (user_hp <= 0) {
76                      cout << "HP가 0 이하가 되었습니다. 실패했습니다." << endl;
77                      cout << "게임을 종료합니다." << endl;
78                      break;
79                  }
80                  cout << "현재 HP : " << user_hp << "  ";
81              }
82          }
```

```
83          else if (user_input == "좌") {
84              // TODO: 왼쪽으로 이동하기
85              user_x -= 1;
86              bool inMap = checkXY(user_x, mapX, user_y, mapY);
87
88              if (inMap == false) {
89                  cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
90                  user_x += 1;
91              }
92              else {
93                  cout << "왼쪽으로 이동합니다." << endl;
94                  displayMap(map, user_x, user_y);
95                  user_hp--;
96                  checkState(map, user_x, user_y);
97                  if (user_hp <= 0) {
98                      cout << "HP가 0 이하가 되었습니다. 실패했습니다." << endl;
99                      cout << "게임을 종료합니다." << endl;
100                     break;
101                 }
102                 cout << "현재 HP : " << user_hp << "  ";
103             }
104         }
```

```
105         else if (user_input == "우") {
106             // TODO: 오른쪽으로 이동하기
107             user_x += 1;
108             bool inMap = checkXY(user_x, mapX, user_y, mapY);
109             if (inMap == false) {
110                 cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
111                 user_x -= 1;
112             }
113             else {
114                 cout << "오른쪽으로 이동합니다." << endl;
115                 displayMap(map, user_x, user_y);
116                 user_hp--;
117                 checkState(map, user_x, user_y);
118                 if (user_hp <= 0) {
119                     cout << "HP가 0 이하가 되었습니다. 실패했습니다."<< endl;
120                     cout << "게임을 종료합니다." << endl;
121                     break;
122                 }
123                 cout << "현재 HP : " << user_hp << "  ";
124             }
125         }
```

```
131         else if (user_input == "종료") {
132             cout << "종료합니다.";
133             break;
134         }
```

A. 입력

상, 하, 좌, 우, 종료

B. 결과

1. 상 일 때 user_ y 값 – 1,

   checkxy 함수 출력값 false 면 다시 user_y 값 +1

   checkxy 함수 출력값 true 면 "위로 한칸 올라갑니다 출력 후 hp--, 만약 hp가 0 이하면 게임 종료

2. 하 일 때 user_ y 값 +1

   checkxy 함수 출력값 false 면 다시 user_y 값 -1

checkxy 함수 출력값 true 면 "아래로 한칸 내려갑니다 출력 후 hp--, 만약 hp가 0 이하면 게임 종료

3. 좌 일 때 user_x 값 – 1

   checkxy 함수 출력값 false 면 다시 user_x 값 +1

   checkxy 함수 출력값 true 면 "왼쪽으로 이동합니다 출력 후 hp--, 만약 hp가 0 이하면 게임 종료

4. 우 일 때 user_x 값 + 1

   checkxy 함수 출력값 false 면 다시 user_x 값 -1

   checkxy 함수 출력값 true 면 "오른쪽으로 이동합니다 출력 후 hp--, 만약 hp가 0 이하면 게임 종료

5. 종료 일 때

   종료합니다. 출력 후 while문 종료

C. 설명

상 하 좌 우 값 입력시 각각 해당되는 좌표를 하나씩 이동시키고 checkxy 함수를 통하여 해당 값이 유효한지 체크, 유효하다면 hp를 한칸 소모하여 유저를 한칸 이동 후 해당 내용 출력. 유효하지 않다면 다시 되돌리고 해당 내용 출력

종료시 종료합니다 출력 후 게임 종료

2. 지도를 입력 시 전체 지도와 함께 현재 위치와 체력을 출력

```
126         else if (user_input == "지도") {
127             // TODO: 지도 보여주기 함수 호출
128             displayMap(map, user_x, user_y);
129             cout << "현재 HP : " << user_hp << "   ";
130         }
```

A. 입력

지도

B. 결과

지도를 보여주는 함수 displayMap을 출력

C. 설명

유저가 지도를 입력한다면 지도를 보여주는 displayMap을 호출하여 유저의 현 위치를 보여준다.

3. 이 중 다른 것을 입력한다면 error 메세지 출력 후 재 입력을 위한 프로세스 진행

```
135         else {
136             cout << "잘못된 입력입니다." << endl;
137             continue;
138         }
139
```

A. 입력

1, 2의 값을 제외한 나머지

B. 결과

잘못된 입력입니다. 출력 후

Continue로 while문 최초로 복귀

C. 설명

지정되지 않은 나머지 값이 들어온다면 잘못된 입력입니다, 를 출력하고 다시 재입력을 시킨다.

ii. 지도 밖으로 이동하라는 명령을 입력시 에러 메시지를 출력

```
if (user_input == "상") {
        // 위로 한 칸 올라가기
        user_y -= 1;
        bool inMap = checkXY(user_x, mapX, user_y, mapY);
        if (inMap == false) {
                cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
                user_y += 1;
        }
```

1. 입력

예시) 유저 입력 "상". - 상하좌우 모두 동일한 흐름

User_y 를 1 뺌

2. 결과

"맵을 벗어났습니다. 다시 돌아갑니다:" 출력

User_y를 1 더함

3. 설명

checkXY 함수를 통하여 유저가 지시 한 값이 맵의 밖을 향하고 있다면 "맵을 벗어났습니다. 다시 돌아갑니다:"를 출력하고, 다시 원래 값으로 되돌린다.

iii. 목적지로 도착하면 성공 메시지를 출력하고 게임을 종료

```
// 목적지에 도달했는지 체크
bool finish = checkGoal(map, user_x, user_y);
if (finish == true) {
        cout << "목적지에 도착했습니다! 축하합니다!" << endl;
        cout << "게임을 종료합니다." << endl;
        break;
}
```

1. 입력

   checkGoal 함수의 값 true or false

2. 결과

   True시 "목적지에 도달했습니다! 축하합니다."

   "게임을 종료합니다." 출력 후 반복문 탈출

   False시 변화 없이 진행

3. 설명

   checkGoal 함수에서 판별한 값이 true가 나온다면 목적지에 유저가 도착한 것 이므로 게임을 종료하고, 반복문을 탈출한다.

iv. 유저의 체력 20이 있고 이동시마다 체력 1 감소, hp가 0 이하가 된다면 실패를 출력하고 게임 종료

```
// 유저 초기 체력을 20으로 정의
int user_hp = 20;
```

```
else {
        cout << "오른쪽으로 이동합니다." << endl;
        displayMap(map, user_x, user_y);
        user_hp--;
        checkState(map, user_x, user_y);
        if (user_hp <= 0) {
                cout << "HP가 0 이하가 되었습니다. 실패했습니다."<< endl;
                cout << "게임을 종료합니다." << endl;
                break;
        }
        cout << "현재 HP : " << user_hp << "  ";
}
```

1. 입력

   User_hp--

2. 결과

User_hp가 0과 작거나 같으면 "HP가 0 이하가 되었습니다. 실패했습니다.", "게임을 종료합니다." 출력 후 게임 종료

3. 설명

전역변수로 user_hp를 20 값을 주어 유저의 체력을 설정했고, 이동시마다 user_hp—로 1씩 줄였으며, 유저 hp 가 0이하면 게임을 종료시키는 멘트와 함께 반복문 탈출로 게임 종료

B. 함수 별 구현 사항 :

i. checkXY : 사용자 캐릭터의 현 위치를 확인하며 적절하면 true, 부적절하면 false를 출력

```
186 // 이동하려는 곳이 유효한 좌표인지 체크하는 함수
187 bool checkXY(int user_x, int mapX, int user_y, int mapY) {
188     bool checkFlag = false;
189     if (user_x >= 0 && user_x < mapX && user_y >= 0 && user_y < mapY) {
190         checkFlag = true;
191     }
192     return checkFlag;
193 }
194
```

1. 입력

Int map[][mapx] = 게임의 전체 지도

Int user_x = 유저의 x 좌표

Int user_y = 유저의 y 좌표

2. 반환값

True

false

3. 결과

True : user x user y 값이 각각 map x, map y 내부에서 동작할 때 출력

false

4. 설명

유저의 변화되는 위치를 전체 맵의 크기 0 <= 유저 < 5 내부에 있는지 확인하여 유저의 이동하려는 값이 유효할지를 체크하여 유효하다면 true를 출력, 아니라면 false를 출력하여 잘못된 동작을 막음

ii. displayMap : 전체 지도와 함께 사용자의 현재 위치를 출력하는 함수

```cpp
// 지도와 사용자 위치 출력하는 함수
void displayMap(int map[][mapX], int user_x, int user_y) {
    for (int i = 0; i < mapY; i++) {
        for (int j = 0; j < mapX; j++) {
            if (i == user_y && j == user_x) {
                cout << " USER |"; // 양 옆 1칸 공백
            }
            else {
                int posState = map[i][j];
                switch (posState) {
                case 0:
                    cout << "      |"; // 6칸 공백
                    break;
                case 1:
                    cout << "아이템|";
                    break;
                case 2:
                    cout << "  적  |"; // 양 옆 2칸 공백
                    break;
                case 3:
                    cout << " 포션 |"; // 양 옆 1칸 공백
                    break;
                case 4:
                    cout << "목적지|";
                    break;
                }
            }
        }
        cout << endl;
        cout << " ------------------------------ " << endl;
    }
}
```

1. 입력

   Int map[][mapx] = 게임의 전체 지도

   Int user_x = 유저의 x 좌표

   Int user_y = 유저의 y 좌표

2. 반환값

   없음

3. 결과

   전체 지도를 출력하고, 사용자의 위치를 출력함

4. 설명

   2차원 배열에 있는 맵을 출력한다.

   출력하다 사용자의 현재 좌표와 동일한 배열의 위치를 발견할 경우 2차원 배열의 값이 아닌 사용자를 출력하여 사용자의 위치를 표현한다.

iii. checkGoal : 유저의 현재 위치가 목적지인지를 확인하고, 목적지 값과 동일하다면 true 값을 출력

```
// 유저의 위치가 목적지인지 체크하는 함수
bool checkGoal(int map[][mapX], int user_x, int user_y) {
    // 목적지 도착하면
    if (map[user_y][user_x] == 4) {
        return true;
    }
    return false;
}
```

1. 입력

   Int map[][mapx] = 게임의 전체 지도

   Int user_x = 유저의 x 좌표

   Int user_y = 유저의 y 좌표

2. 반환값

   True

   False

3. 결과

   사용자의 좌표를 사용한 2차원 배열의 값이 4인 경우 true 값을 출력

   아니면 false 값을 출력

4. 설명

   사용자의 좌표를 사용한 2차원 배열의 값이 4(목적지)일 경우 목적지에 도달했음을 알리는 값인 true를 출력하고, 아니라면 false를 출력하여 계속 게임을 진행한다.

iv. checkState : 유저의 위치에 있는 아이템을 출력하고 해당 값을 hp에 적용

```
// 유저의 위치에 있는 아이템을 출력하고 해당 값을 hp에 적용하는 함수
void checkState(int map[][mapX], int user_x, int user_y) {
    if (map[user_y][user_x] == 2) {
        user_hp -= 2;
        cout << "적이 있습니다. HP가 2 줄어듭니다." << endl;
    }
    else if (map[user_y][user_x] == 3) {
        user_hp += 2;
        cout << "포션이 있습니다. HP가 2 늘어납니다." << endl;
    }
    else if (map[user_y][user_x] == 1) {
        cout << "아이템이 있습니다." << endl;
    }
    return;
}
```

1. 입력

   Int map[][mapx] = 게임의 전체 지도

   Int user_x = 유저의 x 좌표

   Int user_y = 유저의 y 좌표

2. 반환값

   없음

3. 결과

   유저 좌표가 2차원 배열의 값 2일 때 user_hp-=2

   유저 좌표가 2차원 배열의 값 3일 때 user_hp+=2

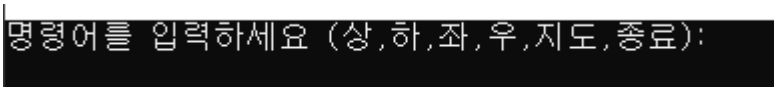
4. 설명

   유저의 좌표를 기반으로 한 2차원 배열 값을 통하여 유저가 적(2)을 만나면 해당 내용을 출력하고 user_hp를 2 감소하며, 포션(3)을 만나면 2증가한다. 아이템(1)은 변화 내용이 없고 해당 내용만 출력한다.

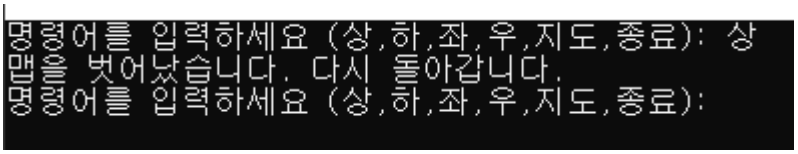
4. 테스트

   A. 기능별 테스트 결과 :

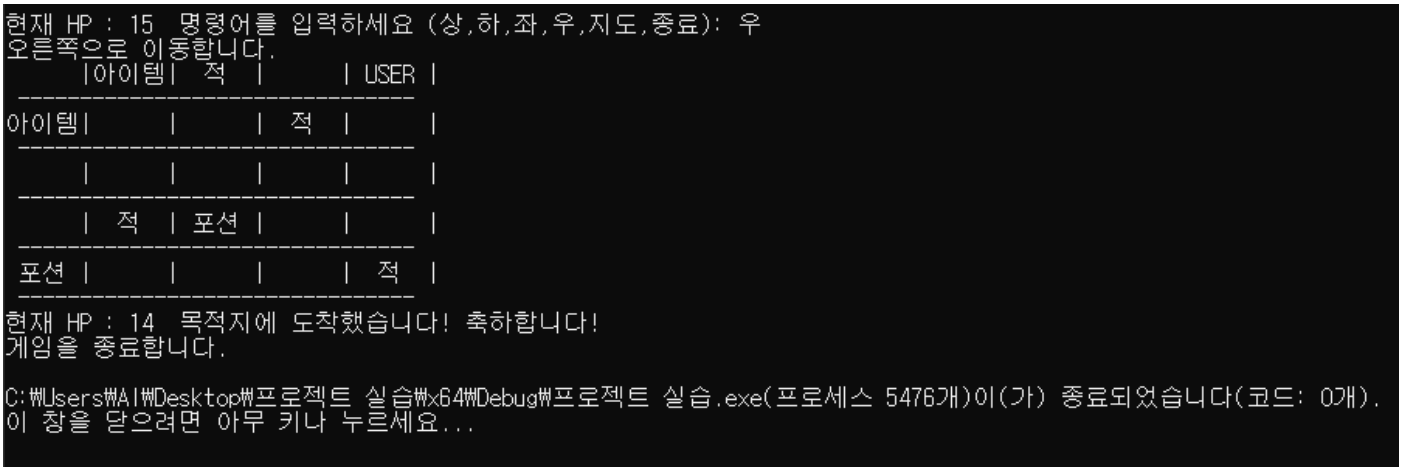
      i. 사용자에게 상하좌우, 지도, 종료 를 입력받기



      ii. 지도 밖으로 이동하라는 명령을 입력시 에러 메시지를 출력

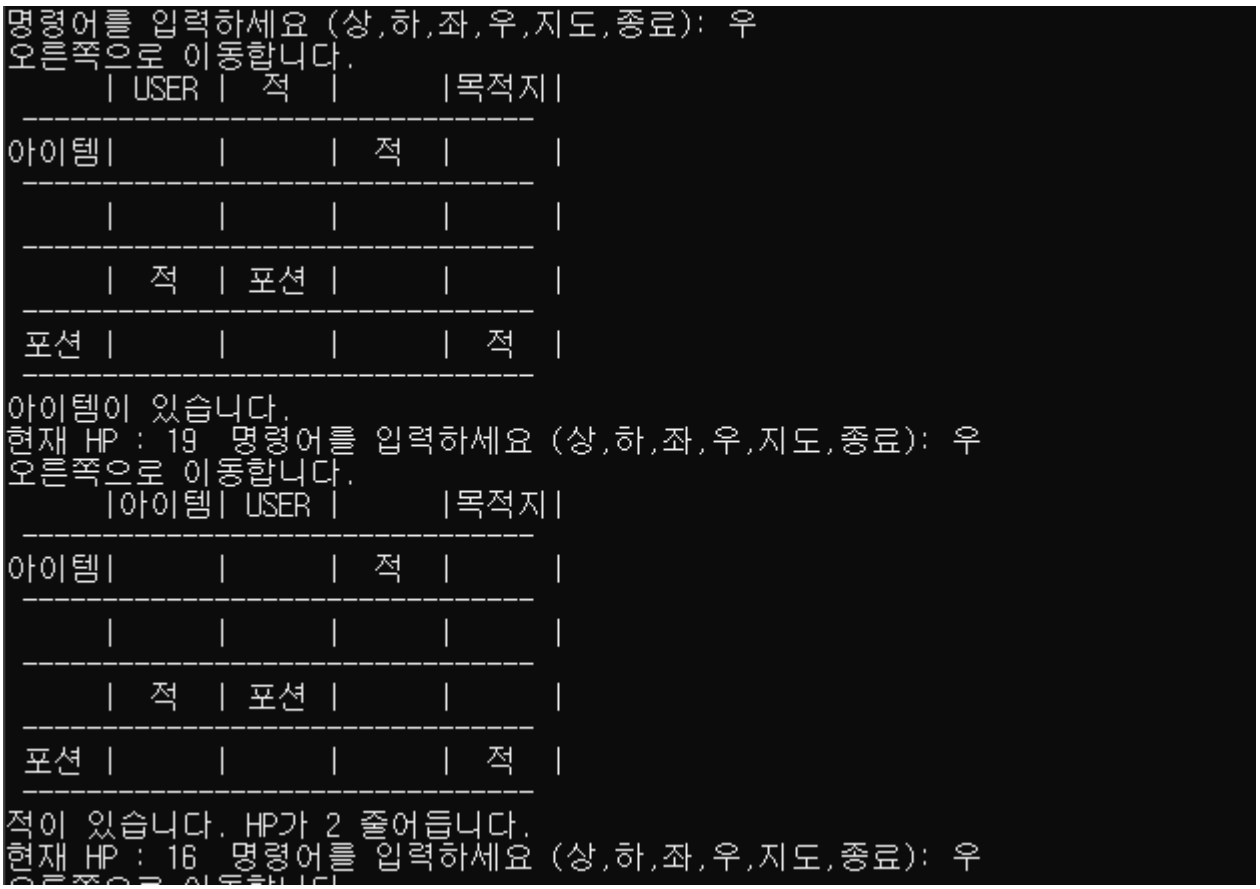


      iii. 목적지로 도착하면 성공 메시지를 출력하고 게임을 종료

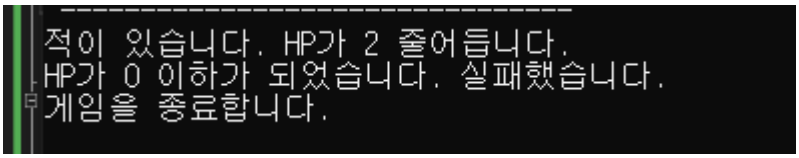


      iv. 유저의 체력 20이 있고 이동 시 체력 1 감소, hp가 0 이하가 된다면 실패를 출력하고 게임 종료

1. 이동 시 체력 1 감소



2. Hp 가 0이 된다면 게임 종료



B. 최종 테스트 스크린샷 :

```
현재 HP : 17  명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 이동합니다.
     |아이템|  적  |      |목적지|
 ------------------------------
아이템|      |      | USER |      |
 ------------------------------
     |      |      |      |      |
 ------------------------------
     |  적  | 포션 |      |      |
 ------------------------------
 포션 |      |      |      |  적  |
 ------------------------------
적이 있습니다. HP가 2 줄어듭니다.
현재 HP : 14  명령어를 입력하세요 (상,하,좌,우,지도,종료): 상
위로 한 칸 올라갑니다.
     |아이템|  적  | USER |목적지|
 ------------------------------
아이템|      |      |  적  |      |
 ------------------------------
     |      |      |      |      |
 ------------------------------
     |  적  | 포션 |      |      |
 ------------------------------
 포션 |      |      |      |  적  |
 ------------------------------
현재 HP : 13  명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 이동합니다.
     |아이템|  적  |      | USER |
 ------------------------------
아이템|      |      |  적  |      |
 ------------------------------
     |      |      |      |      |
 ------------------------------
     |  적  | 포션 |      |      |
 ------------------------------
 포션 |      |      |      |  적  |
 ------------------------------
현재 HP : 12  목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.

C:₩Users₩AI₩Desktop₩프로젝트 실습₩x64₩Debug₩프로젝트 실습.exe(프로세스 12796개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

5. 결과 및 결론

   A. 프로젝트 결과

      기초적인 기능을 구동하고, 체력 시스템이 있으며, 아이템과 포션, 적이 존재하여 유저의 선택에 따라서 목적지에 도달할 수도, 사망할 수도 있는 mud 게임의 구현을 성공.

   B. 느낀 점 : tic-tac-toe를 하다가, 이런 기초적인 mud 게임을 해보니 너무 수월 했습니다.. 시험도 이런 난이도의 코딩이라면, 자신 있게 할 수 있을 것 같은 자신감이 셈 솟습니다!!