

# mud\_game 보고서

203566

서현호

## 보고서 목차

1. 프로젝트 목표 설명
2. mud game 소개
3. 요구사항 목록
- 4 . 요구된 기능 관련 코드 설명
5. 기능 구현 확인
6. 결과 및 피드백

1. 프로젝트의 목표 : 여러 가지 요구된 기능을 수행 할 수 있는 mud game 구현

## 2. mud game 소개

mud game은 user를 목적지 까지 HP를 모두 소비하기 전 목적지에 도착하면 승리하게 되는 게임입니다.

HP : 초기값은 20, 이동할때마다 기본적으로 1씩 감소하며 포션을 만난 경우 2 증가 적을 만난 경우 2 감소

## 3. mud game 구현 시 요구된 기능

1. 지도 밖으로 나가면 에러 메시지 출력
2. '지도'를 입력하면 전체 지도와 함께 현재 위치를 출력
3. 처음 명령문을 받을 때 HP를 같이 출력, 유저의 초기체력 20
4. 사용자가 이동할 때 마다 체력 1 감소
5. HP가 0이 되면 "실패를 출력"
6. 포션을 만났을 때 그에 대한 메시지 출력
7. 적을 만났을 때 그에 대한 메시지 출력

#### 4. 요구사항 관련 코드 설명

(요구사항 관련 따로 정의한 함수들은 관련 코드를 모두 설명한 후 아래에서 설명)

1번,4번 : 지도 밖으로 나가면 에러 메시지 출력,사용자가 이동하면 체력1 감소

```
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;
    }
    else {
        cout << "위로 한 칸 올라갑니다." << endl;
        displayMap(map, user_x, user_y);
        HP--;
    }
}

else if (user_input == "하") {
    // 아래로 한 칸 내려가기
    user_y += 1;
    bool inMap = checkXY(user_x, mapX, user_y, mapY);
    if (inMap == false) {
        cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
        user_y -= 1;
    }
    else {
        cout << "아래로 한 칸 내려갑니다." << endl;
        displayMap(map, user_x, user_y);
        HP--;
    }
}

else if (user_input == "좌") {
    // 왼쪽으로 이동하기
    user_x -= 1;
    bool inMap = checkXY(user_x, mapX, user_y, mapY);

    if (inMap == false) {
        cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
        user_x += 1;
    }
    else {
        cout << "왼쪽으로 이동합니다." << endl;
        displayMap(map, user_x, user_y);
        HP--;
    }
}

else if (user_input == "우") {
    // 오른쪽으로 이동하기
    user_x += 1;
    bool inMap = checkXY(user_x, mapX, user_y, mapY);
    if (inMap == false) {
        cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
        user_x -= 1;
    }
}
```

if문을 사용하여 inMap이 false라면 “맵을 벗어났습니다. 다시 돌아갑니다” 라는 에러 메시지를 출력함. inMap은 함수 checkXY의 값을 입력받는 변수이다. 반복문 while안에 if문을 삽입하여 움직임이 유효할 때 HP가 1씩 감소하도록 하였다.

2번 : “지도”를 입력하면 전체 지도와 함께 현재 위치를 출력

```
    }  
    else {  
        cout << "오른쪽으로 이동합니다." << endl;  
        displayMap(map, user_x, user_y);  
        HP--;  
    }  
}  
else if (user_input == "지도") {  
    // 지도 보여주기 함수 호출  
    displayMap(map, user_x, user_y);  
}  
else if (user_input == "종료") {  
    cout << "종료합니다.";   
    break;  
}  
else {  
    cout << "잘못된 입력입니다." << endl;  
    continue;  
}
```

사용자에게 명령문을 받는 과정에서 “지도”라는 입력을 받으면 user\_input에 저장되고 user\_input에 “지도”가 입력되면 지도를 보여주는 함수를 호출하여 지도를 출력함

3번 : 처음 명령문을 받을 때 HP를 같이 출력하고 유저의 초기체력은 20

```
// 게임 시작  
while (1) { // 사용자에게 계속 입력받기 위해 무한 루프  
  
    // 사용자의 입력을 저장할 변수  
    string user_input = "";  
    cout << "현재 체력은 : " << HP << "\n";  
    cout << "명령어를 입력하세요 (상,하,좌,우,지도,종료): ";  
    cin >> user_input;
```

```
#include <iostream>  
#include <string>  
using namespace std;  
  
const int mapX = 5;  
const int mapY = 5;  
int HP = 20;  
// 게임 구현에 필요한 함수 선언  
bool checkXY(int user_x, int mapX, int user_y, int mapY);  
void displayMap(int map[][mapX], int user_x, int user_y);  
bool checkGoal(int map[][mapX], int user_x, int user_y);  
void checkState(int map[][mapX], int user_x, int user_y);  
// 메인 함수  
int main() {  
    // 게임에 필요한 2차원 배열 정의  
    // 0 = 빈공간, 2 = 적, 3 = 포션, 4 = 목적지  
    int map[mapY][mapX] = { {0, 0, 0, 0, 0},  
                             {0, 0, 0, 0, 0},  
                             {0, 0, 0, 0, 0},  
                             {0, 0, 0, 0, 0},  
                             {0, 0, 0, 0, 0} };
```

게임을 실행하는 while문에 cout를 활용하여 현재 체력을 명령문을 받을 때 항상 같이 출력, HP라는 변수를 20이라는 값으로 초기화시켜서 초기체력을 20으로 만들

5번 : HP가 0이 되면 실패를 출력

```
bool finish = checkGoal(map, user_x, user_y);
if (finish == true) {
    cout << "목표지에 도착했습니다! 축하합니다!" << endl;
    cout << "게임을 종료합니다." << endl;
    break;
}
//HP가 0이되면 종료
if (HP == 0) {
    cout << "실패!! 체력을 모두 소진하였습니다"<<endl;
    break;
}
}
return 0;
```

if문을 사용하여 HP가 0이 되면 '실패!! 체력을 모두 소진하였습니다'를 출력

6번,7번 : 포션,적을 만났을 때 그에 대한 메시지 출력

```
//적을 만난경우와 포션을 만난경우의 유저의 HP 변경을 나타내는 함수
void checkState(int map[][mapX], int user_x, int user_y) {
    //적을 만난 경우
    if (user_y == 0 && user_x == 2 || user_y == 1 && user_x == 3 || user_y == 3 && user_x == 1 ||
        user_y == 4 && user_x == 4) {
        cout << "적을 만났습니다. HP가 2 줄어듭니다" << endl;
        HP -= 2;
    }
    // 포션을 만난경우
    else if (user_y == 3 && user_x == 2 || user_y == 4 && user_x == 0) {
        cout << "포션을 만났습니다. HP가 2 늘어납니다" << endl;
        HP += 2;
    }
}
```

if문을 사용하여 user의 위치가 2차원 배열에서 적과 포션이 있는 칸과 대응된다면 각각 '적을 만났습니다 HP가 2 줄어듭니다', '포션을 만났습니다 HP가 2 늘어납니다'를 출력

함수 displayMap

```
// 사용자 위치가 포함된 지도를 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 2:
                        cout << " 적 "; // 양 옆 2칸 공백
                        break;
                    case 3:
                        cout << " 포션 "; // 양 옆 1칸 공백
                        break;
                    case 4:
                        cout << "목표지";
                        break;
                }
            }
        }
        cout << endl;
        cout << " " << endl;
    }
}
```

반환값이 없는 함수이며, Map[][mapX], user\_x, user\_y 3개의 매개변수를 사용하여 for문과 if문을 사용하여 함수를 활용한다. 가장 바깥의 for문은 mapY보다 작다면 실행되고 1씩 증가한다. 그 안쪽에 있는 for문은 mapX보다 작다면 실행되고 1씩 증가한다. 그 안쪽에 if문을 사용하여 i와 j가 user\_y와 user\_x랑 같다면 'USER '를

출력. 그리고 변수 posState를 선언하고 posState는 배열의 인덱스의 값이 할당된다. switch문을 사용하여 배열 안의 칸에 대응되는 숫자가 posState에 할당되면 위에 각각의 실행해야할 문장들이 실행된다.

#### 함수 checkXY

```
// 이동가능한 칸인지 체크하는 함수
bool checkXY(int user_x, int mapX, int user_y, int mapY) {
    bool checkFlag = false;
    if (user_x >= 0 && user_x < mapX && user_y >= 0 && user_y < mapY) {
        checkFlag = true;
    }
    return checkFlag;
}
```

반환값은 bool 타입으로 반환되고, user\_x, mapX, user\_y, mapY의 매개변수를 활용하는 함수이다. user\_x, user\_y는 유저의 위치가 저장되는 배열의 인덱스이고 만약 유저의 위치가 배열의 인덱스를 벗어난다면 false를 반환하고 반환값은 checkFlag에 저장된다.

#### 함수 checkGoal

```
// 유저의 위치가 목적지에 도달하였는지 체크하는 함수
bool checkGoal(int map[][mapX], int user_x, int user_y) {
    // 목적지에 도착하게 되면 실행
    if (map[user_y][user_x] == 4) {
        return true;
    }
    return false;
}
```

반환값은 bool타입으로 반환되고, map[][mapX], user\_x, user\_y의 매개변수를 활용하는 함수이고, user\_x와 user\_y가 배열에서 4가 들어가있는 칸의 인덱스를 표현하면 목적지에 도착한 것으로 판단하고 그때 true를 반환한다. 아니라면 false를 반환.

#### 함수 checkState

```
return false;
}

//적을 만난경우와 포션을 만난경우의 유저의 HP 변경을 나타내는 함수
void checkState(int map[][mapX], int user_x, int user_y) {
    //적을 만난 경우
    if (user_y == 0 && user_x == 2 || user_y == 1 && user_x == 3 || user_y == 3 && user_x == 1 ||
        user_y == 4 && user_x == 4) {
        cout << "적을 만났습니다. HP가 2 줄어듭니다" << endl;
        HP -= 2;
    }
    // 포션을 만난 경우
    else if (user_y == 3 && user_x == 2 || user_y == 4 && user_x == 0) {
        cout << "포션을 만났습니다. HP가 2 늘어납니다" << endl;
        HP += 2;
    }
}
```

반환값은 없고 매개변수는 위의 그림과 같다. if문을 사용하여 user\_x,user\_y가 포션과 적에 해당하는 인덱스값이 들어있다면 아래의 실행되어야할 문장들이 실행된다.

## 5. 요구된 기능 구현 확인

### 1. 지도 밖으로 나가면 에러 메시지 출력

```
현재 체력은 :20
명령어를 입력하세요 (상,하,좌,우,지도,종료): 지도
USER |      | 적 |      |목적지|
-----|-----|-----|
|      |      |      |      |      |
-----|-----|-----|
|      |      |      |      |      |
-----|-----|-----|
|  적  | 포션 |      |      |      |
-----|-----|-----|
포션 |      |      |      |  적  |
-----|-----|-----|

현재 체력은 :20
명령어를 입력하세요 (상,하,좌,우,지도,종료): 좌
맨을 벗어났습니다. 다시 돌아갑니다.
현재 체력은 :20
명령어를 입력하세요 (상,하,좌,우,지도,종료):
```

### 2. '지도'를 입력하면 전체 지도와 함께 현재 위치를 출력

```
현재 체력은 :20
명령어를 입력하세요 (상,하,좌,우,지도,종료): 지도
USER |      | 적 |      |목적지|
-----|-----|-----|
|      |      |      |      |      |
-----|-----|-----|
|      |      |      |      |      |
-----|-----|-----|
|  적  | 포션 |      |      |      |
-----|-----|-----|
포션 |      |      |      |  적  |
-----|-----|-----|

현재 체력은 :20
명령어를 입력하세요 (상,하,좌,우,지도,종료):
```

### 3. 처음 명령문을 받을 때 HP를 같이 출력, 유저의 초기체력 20

```
C:\Users\203566\Desktop\cpp\Project1\wx64\Debug\Project1.exe
현재 체력은 :20
명령어를 입력하세요 (상,하,좌,우,지도,종료):
```



#### 4. 사용자가 이동할 때 마다 체력 1 감소

```
현재 체력은 :20
명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 이동합니다.
| USER | 적 | 목적지 |
-----
|      |   |       |
-----
|      |   |       |
-----
|  적  | 포션 |       |
-----
포션 |      |       |  적  |
-----
현재 체력은 :19
명령어를 입력하세요 (상,하,좌,우,지도,종료):
```

#### 5. HP가 0이 되면 실패를 출력

```
현재 체력은 :3
명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 이동합니다.
|      | USER | 목적지 |
-----
|      |      |  적   |
-----
|      |      |       |
-----
|  적  | 포션 |       |
-----
포션 |      |       |  적  |
-----
적을 만났습니다. HP가 2 줄어듭니다
실패!! 체력을 모두 소진하였습니다

C:\Users\203566\Desktop\cpp\Project1\x64\Debug\Project1.exe
이 창을 닫으려면 아무 키나 누르세요...
```

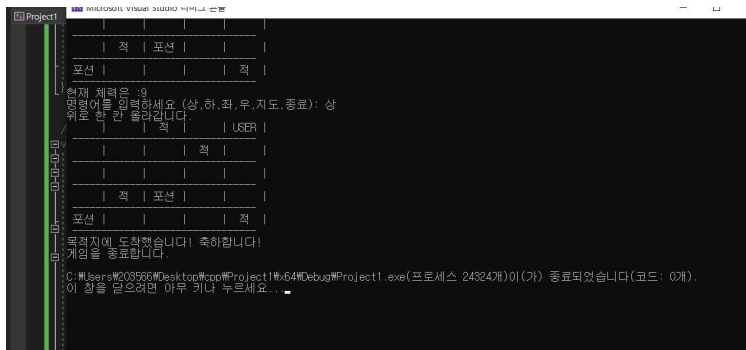
6. 포션을 만났을 때 그에 대한 메시지 출력

```
아래로 한 칸 내려갑니다.
|      |      |      |      |목적지|
-----
|      |      |      |      |      |
-----
|      |      |USER |      |      |
-----
|      |      |포션 |      |      |
-----
포션 |      |      |      |      |      |      |      |      |      |      |      |
-----
현재 체력은 :14
몬스터를 공격하세요 (상,하,좌,우,지도,종료): 하
아래로 한 칸 내려갑니다.
|      |      |      |      |목적지|
-----
|      |      |      |      |      |
-----
|      |      |      |      |      |
-----
|      |      |USER |      |      |
-----
포션 |      |      |      |      |      |      |      |      |      |      |      |
-----
포션을 만났습니다. HP가 2 늘어납니다
현재 체력은 :15
몬스터를 공격하세요 (상,하,좌,우,지도,종료):
```

### 7. 적을 만났을 때 그에 대한 메시지 출력

[illegible]

## 6. 결과 및 피드백



(최종적으로 결과를 완수하였을 때의 화면)

피드백 :

최종적으로 게임을 완료하였고 완벽히 실행되는 모습을 보임.

함수의 활용에 대하여 더 깊은 이해와 활용 능력을 함양하게 되었음.

지역 변수와 전역 변수에 대한 흐름에 관한 더 많은 이해 필요함.