

# **MUD·GAME·REPORT**

**학번 : 213996**

**이름 : 한승한**

## 목표

1. 주어진 코드를 가지고 MUD GAME을 완성시킨다.
2. 반복되는 구문을 보완한다.

## 요구사항

1. 유저는 체력 20을 가지고 시작
  2. 이동시 체력 1씩 감소
  3. 명령문을 입력 받을 때마다 HP 출력
  4. HP가 0이 되면 "실패"를 출력하고 프로그램 종료
  5. 아이템 / 적 / 포션을 만났을 때 그에 대한 메시지를 출력
    - 5 - 1) 적과 조우할 경우 HP가 2 줄어 들고 그에 대한 메시지 출력
    - 5 - 2) 포션을 획득할 경우 HP가 2 늘어나고 그에 대한 메시지 출력
- ※ 적이나 포션은 사라지지 않음을 전제 ※

## **코드·별·기능**

```
// 메인 함수
int main() {
    // 0은 빈 공간, 1은 아이템, 2는 적, 3은 포션, 4는 목적지
    int map[mapY][mapX] = { {0, 1, 2, 0, 4},
                             {1, 0, 0, 2, 0},
                             {0, 0, 0, 0, 0},
                             {0, 2, 3, 0, 0},
                             {3, 0, 0, 0, 2} };

    // 유저의 위치를 저장할 변수
    int user_x = 0; // 가로 번호
    int user_y = 0; // 세로 번호

    // 유저의 체력을 저장할 변수 : 20으로 시작
    int user_hp = 20;
}
```

<스크린샷 (上)>

```
displayMap(map, user_x, user_y);

// 게임 시작
while (1) { // 사용자에게 계속 입력받기 위해 무한 루프

    // 사용자의 입력을 저장할 변수
    string user_input = "";
    cout << "현재 HP: " << user_hp << " ";
    cout << "영어로 명령어를 입력하세요 [상(w), 하(s), 좌(a), 우(d), 지도(map), 종료(exit)]: ";
    cin >> user_input;
}
```

<스크린샷 (下)>

## 입력 및 설명 (上)

int map[mapY][mapX] = 전체 지도를 먼저 초기화. 상세 설명은 주석

int user\_x = 유저가 움직일 x좌표

int user\_y = 유저가 움직일 y좌표

int user\_hp = 유저의 HP : 20

## 입력 및 설명 (下)

displayMap = 처음 게임을 시작할 때 지도가 보이게 설정

string user\_input = 유저의 입력을 저장할 변수

cout = 안내문 출력

cin = 유저의 입력을 변수로 저장

```
if (user_input == "w") {  
    // 위로 한 칸 올라가기  
    if (checkXY(user_x, mapX, user_y - 1, mapY)) { // 미리 움직였을 때 map 범위 안에 존재할 경우  
        user_y -= 1;  
        cout << "위로 한 칸 올라갑니다." << endl;  
        displayMap(map, user_x, user_y);  
        user_hp = Print(map, user_x, user_y, user_hp); // 목적지 별 메시지 출력 + 체력 감소  
    }  
}  
else if (user_input == "s") {  
    // TODO: 아래로 한 칸 내려가기  
    if (checkXY(user_x, mapX, user_y + 1, mapY)) { // 미리 움직였을 때 map 범위 안에 존재할 경우  
        user_y += 1;  
        cout << "아래로 한 칸 내려갑니다." << endl;  
        displayMap(map, user_x, user_y);  
        user_hp = Print(map, user_x, user_y, user_hp); // 목적지 별 메시지 출력 + 체력 감소  
    }  
}  
else if (user_input == "a") {  
    // TODO: 왼쪽으로 이동하기  
    if (checkXY(user_x - 1, mapX, user_y, mapY)) { // 미리 움직였을 때 map 범위 안에 존재할 경우  
        user_x -= 1;  
        cout << "왼쪽으로 이동합니다." << endl;  
        displayMap(map, user_x, user_y);  
        user_hp = Print(map, user_x, user_y, user_hp); // 목적지 별 메시지 출력 + 체력 감소  
    }  
}  
else if (user_input == "d") {  
    // TODO: 오른쪽으로 이동하기  
    if (checkXY(user_x + 1, mapX, user_y, mapY)) { // 미리 움직였을 때 map 범위 안에 존재할 경우  
        user_x += 1;  
        cout << "오른쪽으로 이동합니다." << endl;  
        displayMap(map, user_x, user_y);  
        user_hp = Print(map, user_x, user_y, user_hp); // 목적지 별 메시지 출력 + 체력 감소  
    }  
}
```

<스크린샷 (上)>

## 입력 및 설명 (上)

if or else if (user\_input == 각 움직임에 따른 if문 처리)

\* 원인은 모르겠지만 한글 입력에 작동이 불가하여 영문으로 바꿈.

\* 방향키에 의거 w, a, s, d로 입력받음

if (checkXY = 만약 허용된 범위 내에서 움직인다면 실행)

\* 입력받은 움직임으로 먼저 이동시킨 후 좌표를 검사하여 유효한지 체크

user\_y -= 1 유저의 **Y좌표**를 1칸 줄임 (좌표상 上로 이동)

user\_y += 1 유저의 **Y좌표**를 1칸 늘림 (좌표상 下로 이동)

user\_x -= 1 유저의 **X좌표**를 1칸 줄임 (좌표상 左로 이동)

user\_x += 1 유저의 **X좌표**를 1칸 늘림 (좌표상 右로 이동)

cout = 메시지 출력

displayMap = 지도 출력

user\_hp = Print 이동하고 남은 체력을 변수에 저장

```

else if (user_input == "map") {
    // TODO: 지도 보여주기 함수 호출
    displayMap(map, user_x, user_y);
}
else if (user_input == "exit") {
    cout << "종료합니다.";
    break;
}
else {
    cout << "잘못된 입력입니다." << endl;
    continue;
}

```

## 입력 및 설명

else if (user\_input = 유저 입력에 따른 이벤트 처리)

\* map을 입력할 시 지도를 보여주고, exit를 입력할 시 프로그램을 종료한다.

else = 위의 명령어가 아닌 입력이 접근할 시 종료한다.

cout = 잘못된 입력임을 알려주는 경고문

continue = 잘못된 입력시 처음으로 돌아가 다시 입력하게 도와준다.

```

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

// HP_Checker : HP가 0이 되면 실패를 출력하고 종료
if (HP_Checker(user_hp))
    break;
}
return 0;

```

## 입력 및 설명

bool finish = checkGoal 저장변수에 bool값을 저장한다.

\* checkGoal은 목적지에 도착했는지 검사하는 함수

\* 목적지에 도착시 true값을 반환한다. 그렇지 않다면 false값을 반환한다.

if (finish == true) 저장변수의 값이 true가 된다면 실행한다.

cout = 안내문구를 출력한다.

break = 현재 무한히 실행되고 있는 while문을 벗어나므로 프로그램을 종료.

if (HP\_Checker = 현재 남아있는 체력이 0이 됐을 경우 true값을 반환한다.)

break = 위 설명과 동일하다.

return 0 = 프로그램을 종료한다.



```

// Print 함수 : 각 상황에 맞는 메세지 출력 + 남은 HP를 반환
int Print(int map[][mapX], int user_x, int user_y, int user_hp) {
    user_hp -= 1;
    switch(map[user_y][user_x]) { // col, row 순서
        case 0: // 0은 빈 공간, 1은 아이템, 2는 적, 3은 포션, 4는 목적지
            cout << "아무것도 없었습니다." << endl;
            break;
        case 1:
            cout << "아이템을 발견했습니다." << endl;
            break;
        case 2:
            cout << "적과 조우했습니다." << " " << "체력이 '2' 감소합니다." << endl;
            cout << "현재 HP: " << user_hp;
            user_hp -= 2; // 체력 2 감소
            cout << " -> " << user_hp << endl;
            break;
        case 3:
            cout << "포션을 발견했습니다." << " " << "체력을 '2' 회복합니다." << endl;
            cout << "현재 HP: " << user_hp;
            user_hp += 2; // 체력 2 증가
            cout << " -> " << user_hp << endl;
            break;
    }
    return user_hp; // 남은 체력 반환
}

```

## 입력 및 설명

int Print ()= 함수를 정의한다.

\* 반환값이 int형인 user\_hp임으로 int형으로 선언한다.

user\_hp -= 1;

\* 이 함수가 실행될 때마다 체력이 1씩 감소한다. (이동 시 감소를 표현)

switch(map[user\_y][user\_x]) = 현재 유저의 위치를 switch문에 사용한다.

\* 1차원 배열이 X축(row)이며 2차원 배열이 Y축(col)이다.

case 0: = 현재 유저의 위치가 0에 위치한다면 실행된다. (빈 공간)

cout = 안내문구를 출력한다.

break = switch문을 빠져나와 user\_hp를 반환한다.

case 1: = 위와 동일하다. (아이템)

cout, break = 위와 동일하다.

case 2: = 위와 동일하다. (적)

cout = 현재의 HP 상태와 적을 만나 감소한 HP 상태를 보여준다.

user\_hp -= 2 적을 만나 HP가 2 감소한다.

break = 위와 동일하다.

case 3: = 위와 동일하다. (포션)

cout = 현재의 HP 상태와 회복 후의 HP 상태를 보여준다.

user\_hp += 2 포션을 획득하여 HP가 2 증가한다.

break = 위와 동일하다.

```
// HP가 0이 되면 종료를 출력하는 함수
bool HP_Checker(int user_hp) {
    bool b = false;
    if (user_hp <= 0) {
        cout << "실패";
        b = true;
    }
    return b;
}
```

## 입력 및 설명

bool HP\_Checker() = 함수를 정의한다.

\* bool의 값을 반환하기 때문에 bool형으로 정의한다.

bool b = false 기본 반환값을 false로 반환한다.

if (user\_hp <= 0) 만약 유저 체력이 0보다 작거나 같다면 실행

cout = "실패"라는 문구를 출력한다.

b = true 기본 반환값을 true로 반환한다.

return b = b함수의 bool값을 반환한다.

```

// 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;
    }
}

```

## 입력 및 설명

void displayMap() = 지도를 출력하는 함수를 정의한다.

for ( i < mapY) = 인자가 mapY - 1까지 반복된다.

for( j < mapX) = 인자가 mapX -1까지 반복된다.

if (i == user\_y && j == user\_x)

\* 현재 위치한 좌표에 USER가 출력된다.

else = 현재 위치하지 않은 좌표에서 실행된다.

int posState = map[i][j]

\* 맵의 모든 좌표값(0~4)을 posState에 저장한다.

switch (posState) = posState값을 사용한다.

case 0: posState값이 0일때 실행된다.

cout = 칸을 공백으로 출력한다.

break = switch문을 벗어난다.

case 1: posState값이 1일때 실행된다.

cout = 아이템 칸을 출력한다.

break = switch문을 벗어난다.

case 2: posState값이 2일때 실행된다.

cout = 적 칸을 출력한다.

break = switch문을 벗어난다.

case 3: posState값이 3일때 실행된다.

cout = 포션 칸을 출력한다.

break = switch문을 벗어난다.

case 4: posState값이 4일때 실행된다.

cout = 목적지 칸을 출력한다.

break = switch문을 벗어난다.

cout = 줄바꿈을 한 번 출력한다.

cout = 절취선을 출력한다.

```
// 이동하려는 곳이 유효한 좌표인지 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;
    }
    else {
        cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
    }
    return checkFlag;
}
```

## 입력 및 설명

bool checkXY() = 유효한 좌표인지 검사하는 함수 선언

\* bool 값을 반환하므로 bool형으로 선언

bool checkFlag = false; 기본값을 false로 선언

if ( 유효한 범위면 실행 )

\* 유저의 X좌표가 0보다 커야하며 맵의 X좌표보다 작아야 한다.

\* 유저의 Y좌표가 0보다 커야하며 맵의 Y좌표보다 작아야 한다.

checkFlag = true 유효한 범위면 true값 반환

else ( 유효한 범위가 아니면 실행 )

cout = 경고문 출력

return checkFlag = checkFlag의 bool값을 반환

```

// 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;
}

```

## 입력 및 설명

bool checkGoal() = 현재 위치가 목적지인지 체크하는 함수 선언

if (map[user\_y][user\_x] == 4) = 현재 위치가 4(목적지)면 실행

return true = true값 반환

return false = 목적지가 아니면 false값 반환

**TEST**

```

USER |아이템| 적 | |목적지|
-----
아이템| | | |적 | |
-----
| | | | | |
-----
| |적 |포션 | | |
-----
포션 | | | |적 |
-----
현재 HP: 20 영어로 명령어를 입력하세요 [상(w), 하(s), 좌(a), 우(d), 지도(map), 종료(exit)]: 

```

```

현재 HP: 20 영어로 명령어를 입력하세요 [상(w), 하(s), 좌(a), 우(d), 지도(map), 종료(exit)]: d
오른쪽으로 이동합니다.
      | USER |   직   |   목적지   |
-----|-----|-----|
아이템|   |   |   |   |
-----|-----|-----|
      |   |   |   |   |
-----|-----|-----|
      |   |   |   |   |
-----|-----|-----|
      |   |   |   |   |
-----|-----|-----|
      |   |   |   |   |
-----|-----|-----|
      |   |   |   |   |
-----|-----|-----|
아이템을 발견했습니다.
현재 HP: 19 영어로 명령어를 입력하세요 [상(w), 하(s), 좌(a), 우(d), 지도(map), 종료(exit)]:

```

```

현재 HP: 1 영어로 명령어를 입력하세요 [상(w), 하(s), 좌(a), 우(d), 지도(map), 종료(exit)]: w
위로 한 칸 올라갑니다.
|아이템|  적  |      |목적지|
-----
아이템|      |      | USER |      |
-----
|      |      |      |      |      |
-----
|  적  | 포션 |      |      |      |
-----
포션|      |      |      |  적  |
-----
적과 조우했습니다. 체력이 '2'감소합니다.
현재 HP: 0 -> -2
실패
PS C:\CPP2409>

```





## 최종본

```

USER |아이템|  적  |      |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션  |      |      |      |  적  |
-----
현재 HP: 20  영어로 명령어를 입력하세요 [상(w), 하(s), 좌(a), 우(d), 지도(map), 종료(exit)]: d
오른쪽으로 이동합니다.
      | USER |  적  |      |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션  |      |      |      |  적  |
-----
아이템을 발견했습니다.
현재 HP: 19  영어로 명령어를 입력하세요 [상(w), 하(s), 좌(a), 우(d), 지도(map), 종료(exit)]: d
오른쪽으로 이동합니다.
      |아이템| USER |      |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션  |      |      |      |  적  |
-----
적과 조우했습니다.  체력이 '2'감소합니다.
현재 HP: 18 -> 16
현재 HP: 16  영어로 명령어를 입력하세요 [상(w), 하(s), 좌(a), 우(d), 지도(map), 종료(exit)]: s
아래로 한 칸 내려갑니다.

```

현재 HP: 16 영어로 명령어를 입력하세요 [상(w), 하(s), 좌(a), 우(d), 지도(map), 종료(exit)]: s  
아래로 한 칸 내려갑니다.

아이템	적	목적지
-----	---	-----

아이템	USER	적
-----	------	---

|--|--|--|

	적	포션
--	---	----

포션		적
----	--	---

아무것도 없었습니다.

현재 HP: 15 영어로 명령어를 입력하세요 [상(w), 하(s), 좌(a), 우(d), 지도(map), 종료(exit)]: s  
아래로 한 칸 내려갑니다.

아이템	적	목적지
-----	---	-----

아이템		적
-----	--	---

	USER	
--	------	--

	적	포션
--	---	----

포션		적
----	--	---

아무것도 없었습니다.

현재 HP: 14 영어로 명령어를 입력하세요 [상(w), 하(s), 좌(a), 우(d), 지도(map), 종료(exit)]: s  
아래로 한 칸 내려갑니다.

아이템	적	목적지
-----	---	-----

아이템		적
-----	--	---

|--|--|--|

	적	USER
--	---	------

포션		적
----	--	---

포션을 발견했습니다. 체력을 '2'회복합니다.

현재 HP: 13 -> 15

현재 HP: 15 영어로 명령어를 입력하세요 [상(w), 하(s), 좌(a), 우(d), 지도(map), 종료(exit)]: d  
오른쪽으로 이동합니다.

아이템		적		목적지	
-----					
아이템				적	
-----					
-----					
		적		포션	
-----					
포션					
-----					

아무것도 없었습니다.

현재 HP: 14 영어로 명령어를 입력하세요 [상(w), 하(s), 좌(a), 우(d), 지도(map), 종료(exit)]: d  
오른쪽으로 이동합니다.

아이템		적		목적지	
-----					
아이템				적	
-----					
-----					
		적		포션	
-----					
포션					
-----					

아무것도 없었습니다.

현재 HP: 13 영어로 명령어를 입력하세요 [상(w), 하(s), 좌(a), 우(d), 지도(map), 종료(exit)]: w  
위로 한 칸 올라갑니다.

아이템		적		목적지	
-----					
아이템				적	
-----					
-----					
		적		포션	
-----					
포션					
-----					

아무것도 없었습니다.

```

현재 HP: 12 영어로 명령어를 입력하세요 [상(w), 하(s), 좌(a), 우(d), 지도(map), 종료(exit)]: w
위로 한 칸 올라갑니다.
  |아이템|  적  |      |목적지|
-----
아이템|      |      |  적  | USER |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
아무것도 없었습니다.
현재 HP: 11 영어로 명령어를 입력하세요 [상(w), 하(s), 좌(a), 우(d), 지도(map), 종료(exit)]: w
위로 한 칸 올라갑니다.
  |아이템|  적  |      | USER |
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.

```

## 결과 및 결론

1. 함수화 된 매개변수에 2차배열을 넣을 때 가변된 값을 넣을 수 없어서 당황했다.
2. 체력이 0이 되면 실행되는 함수를 구현할 때 hp==0으로 구현했었는데 오류가 생겼다.
  - \* 체력이 1인 상황에서 적 조우시 음수값으로 변하며 불사의 인간이 돼버렸다.
3. 한글 입력이 먹히지 않아 처음 10분동안 헛스윙을 했다.
4. 결론적으로 잘 기능하는 것 같아 만족스러웠다.