



# MUD\_GAME .CPP

## 프로젝트 요약

2023-10-23	프로젝트 이름	작성자
전남대학교	Mud_game.cpp	한동호

## 상황 요약

10 월 23 일 교수님께서 Mud 게임 관련 실습을 진행, 이에 따른 보고서 작성.  
목표 – 요구사항을 반영한 간단한 Mud 게임 구현

## 요구사항(기능 계획 포함)

1. 사용자 요구사항: 유저가 상하좌우로만 이동하며 목적지에 도착하는 게임

### 2. 기능 요구사항

#### ① 사용자에게 상, 하, 좌, 우, 지도, 종료 중 하나를 입력받기

- 상/하/좌/우 입력시 해당 방향으로 이동 후 지도 출력
- “지도”를 입력하면 전체 지도와 함께 현재 위치를 출력
- 이 중 다른 것을 입력하면 에러 메시지 출력 후 재 입력 요청

#### ② 지도 밖으로 나가게 되면 에러 메시지 출력

#### ③ 목적지에 도착하면 “성공을 출력하고 종료

+ <추가 요구사항>

- ① 유저는 체력 20 을 가지고 게임 시작
  - ② 사용자가 이동할 때 마다 사용자 체력 1 씩 감소
  - ③ 처음 명령문을 입력 받을 때 마다 HP 함께 출력
  - ④ HP 가 0 이 되면 “실패”를 출력하고 종료
  - ⑤ 무기/갑옷, 포션, 적을 만났을 때 그에 대한 메시지를 출력
- 예) {X}가 있습니다.
  - 적을 만날 경우 HP 가 2 가 줄어들고 그에 대한 추가 메시지 출력

## 함수계획

메인 함수: 사용자에게 값을 계속 입력받고, 그에 대한 함수 호출

지도와 현재 위치 출력 함수: `displayMap()`

사용자 위치 체크 함수: `checkXY()`

목적지에 도착 체크 함수: `checkGoal()`

아이템, 적, 포션을 만났을 때 체력변화 및 메시지 출력함수: `checkState()`

① 사용자에게 상, 하, 좌, 우, 지도, 종료 중 하나를 입력받기

And

② 지도 밖으로 나가게 되면 에러 메시지 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;
    }
    else {
        cout << "위로 한 칸 올라갑니다." << endl;
        displayMap(map, user_x, user_y);
    }
}
else if (user_input == "하") {
    // TODO: 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);
    }
}
else if (user_input == "좌") {
    // TODO: 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);
    }
}
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;
        }
        else {
            cout << "오른쪽으로 이동합니다." << endl;
            displayMap(map, user_x, user_y);
        }
    }
    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;
    }
}

```

### ③ 목적지에 도착하면 “성공을 출력하고 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;
}

```

## + <추가 요구사항>

### ① 유저는 체력 20 을 가지고 게임 시작

```

// 유저의 체력을 저장할 변수
int userhealth = 20;

```

### ② 사용자가 이동할 때 마다 사용자 체력 1 씩 감소

```

// 횟수 및 체력 체크
usermove++;
userhealth = 20 - usermove;

```

### ③ 처음 명령문을 입력 받을 때 마다 HP 함께 출력

```

cout << "현재 HP : " << userhealth;

```

#### ④ HP 가 0 이 되면 “실패”를 출력하고 종료

```
// 목적지에 가지 못하고 HP가 0일 경우도 체크
else if (userhealth == 0) {
    cout << "HP가 0 이하가 되었습니다. 실패했습니다. " << endl;
    cout << "게임을 종료합니다." << endl;
    break;
}
```

#### ⑤ 무기/갑옷, 포션, 적을 만났을 때 그에 대한 메시지를 출력

//무기 및 갑옷, 포션, 적을 만났을 때 그에 대한 메시지를 출력하는 함수

```
void checkState(int map[][mapX], int user_x, int user_y)
{
    if (map[user_y][user_x] == 2)
    {
        cout << "적이 있습니다. HP가 2줄어듭니다.\n ";
        usermove += 2;
    }
    else if (map[user_y][user_x] == 3)
    {
        cout << "포션이 있습니다. HP가 2늘어납니다.\n ";
        usermove -= 2;
    }
    else if (map[user_y][user_x] == 1)
    {
        cout << "아이템이 있습니다.\n";
    }
}
}
//
```

## <함수구현>

메인 함수: 사용자에게 값을 계속 입력받고, 그에 대한 함수 호출

```
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
    int userhealth = 20;
```

```

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

    // 사용자의 입력을 저장할 변수
    string user_input = "";

    cout << "현재 HP : " << userhealth;
    cout << " 명령어를 입력하세요 (상,하,좌,우,지도,종료): ";
    cin >> user_input;

    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);
        }
    }
    else if (user_input == "하") {
        // TODO: 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);
        }
    }
    else if (user_input == "좌") {
        // TODO: 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);
        }
    }
}

```

```

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;
    }
    else {
        cout << "오른쪽으로 이동합니다." << endl;
        displayMap(map, user_x, user_y);
    }
}
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;
}

//변수 고려한 체력 체크
checkState(map, user_x, user_y);

// 횟수 및 체력 체크
usermove++;
userhealth = 20 - usermove;

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

} // 목적지에 가지 못하고 HP가 0일 경우도 체크
else if (userhealth == 0) {
    cout << "HP가 0 이하가 되었습니다. 실패했습니다. " << endl;
    cout << "게임을 종료합니다." << endl;
    break;
}
}
return 0;
}

```

// 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;
    }
}

```

#### 지도와 현재 위치 출력 함수: displayMap()

```

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;
}
}

```

#### 사용자 위치 체크 함수: checkXY()

// 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;
    }
    return checkFlag;
}

```

#### 목적지에 도착 체크 함수: checkGoal()

// 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;
}

```

#### 아이템, 적, 포션을 만났을 때 체력변화 및 메시지 출력함수: checkState()

//무기 및 갑옷, 포션, 적을 만났을 때 그에 대한 메시지를 출력하는 함수

```

void checkState(int map[][mapX], int user_x, int user_y)
{
    if (map[user_y][user_x] == 2)
    {
        cout << "적이 있습니다. HP가 2줄어듭니다.\n ";
        usermove += 2;
    }
    else if (map[user_y][user_x] == 3)
    {
        cout << "포션이 있습니다. HP가 2늘어납니다.\n ";
        usermove -= 2;
    }
    else if (map[user_y][user_x] == 1)
    {
        cout << "아이템이 있습니다.\n";
    }
}

```

## 테스트 (기능별)

- ① 사용자에게 상, 하, 좌, 우, 지도, 종료 중 하나를 입력받기

명령어를 입력하세요 (상, 하, 좌, 우, 지도, 종료):

- ② 지도 밖으로 나가게 되면 에러 메시지 출력

맵을 벗어났습니다. 다시 돌아갑니다.

- ③ 목적지에 도착하면 “성공을 출력하고 종료

목적지에 도착했습니다! 축하합니다!  
게임을 종료합니다.

## + <추가 요구사항>

- ① 유저는 체력 20 을 가지고 게임 시작

현재 HP : 20

- ② 사용자가 이동할 때 마다 사용자 체력 1 씩 감소 + ③ 처음 명령문을 입력 받을 때 마다 HP 함께 출력

현재 HP : 20    명령어를 입력하세요 (상, 하, 좌, 우, 지도, 종료): 하  
아래로 한 칸 내려갑니다.

|아이템|    적    |    |목적지|

-----  
USER |    |    |    적    |    |

-----  
|    |    |    |    |    |

-----  
|    적    | 포션    |    |    |    |

-----  
포션    |    |    |    |    적    |

-----  
아이템이 있습니다.

현재 HP : 19    명령어를 입력하세요 (상, 하, 좌, 우, 지도, 종료): 하

- ④ HP 가 0 이 되면 “실패”를 출력하고 종료

```

-----
적이 있습니다. HP가 2줄어듭니다.
HP가 0 이하가 되었습니다. 실패했습니다.
게임을 종료합니다.

```

⑤ 무기/갑옷, 포션, 적을 만났을 때 그에 대한 메시지를 출력

```

-----
아이템이 있습니다.
현재 HP : 10 명령어를 입력하세요
-----
포션이 있습니다. HP가 2늘어납니다.
현재 HP : 12 명령어를 입력하세요
-----
적이 있습니다. HP가 2줄어듭니다.

```

최종 테스트 스크린샷

```

현재 HP : 15 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 이동합니다.
  |아이템|  적  |      | USER |
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.

```

## 결과 및 결론

### 프로젝트 결과

- ⇒ Mud 게임을 구현 성공

### 느낀 점

- ⇒ 시험기간에 실습을 하니 수업시간 이후로 미뤄선 안되겠다는 사명감이 생겼다  
덕분에 수업시간안에 무사히 해냈다!

