

# Mud game 보고서

233958 김강문

## 1. 서론

1. **프로젝트 목적 및 배경:** 단순 입출력, 반복문, 조건문, 함수화 등의 내용을 이용해 중간고사 이전의 내용을 활용할 수 있는지 확인하기 위한 실습
2. **목표:** 간단한 Mud 게임 구현하기

## 2. 요구사항

1. **사용자 요구사항:** 유저가 상하좌우로 이동하며 HP가 0이 될 때까지 목적지에 도착하는 게임
2. **기능 계획**
  - 유저의 기본 체력을 20으로 설정하고 한 칸을 이동할 때 마다 체력 1씩 감소
  - 명령문을 입력 받을 때 마다 HP도 출력
  - 상하좌우 이동 후 지도 사용자의 현재 위치 출력
  - 지도를 입력하면 지도와 사용자의 현재 위치 출력
  - 종료를 입력하면 프로그램 종료
  - 그 외 다른 것을 입력하면 에러 메시지 출력 후 재입력 요청
  - 지도 밖으로 나가게 되면 에러메시지 출력 후 재입력 요청
  - 아이템을 만날 경우 포션의 위치로 이동할 것인지 묻고 답변에 따라 이동하거나 이동하지 않고 그에 대한 추가 메시지 출력
  - 적을 만날 경우 HP가 2 줄어들고 그에 대한 메시지 출력
  - 포션을 만날 경우 HP가 2 늘어나고 그에 대한 메시지 출력
  - 목적지에 도착하면 "성공"을 출력하고 종료 (목적지에 도착했고 체력이 0인 경우도 성공으로 간주)
  - 체력이 0이 되면 "실패"를 출력하고 종료

### 3. 함수 계획

- 메인 함수: 사용자에게 무한루프를 통해 값을 입력받고, 그에 동작 및 함수 호출
- bool checkXY(int user\_x, int mapX, int user\_y, int mapY) 함수를 생성해 이동하려는 곳이 유효한 좌표인지 확인한다.
- void displayMap(int map[mapY][mapX], int user\_x, int user\_y) 함수를 생성해 사용자의 위치와 지도를 출력한다.
- bool checkGoal(int map[mapY][mapX], int user\_x, int user\_y) 함수를 생성해 사용자의 위치가 목적지인지 확인하고 프로그램을 종료한다.
- void checkState(int map[mapY][mapX], int& user\_x, int& user\_y, int& hp) 함수를 생성해 아이템, 적, 포션을 만났을 때 그에 대한 기능을 동작하고 메시지를 출력한다.

## 3. 설계 및 구현

### 1. 기능 별 구현 사항: (요구사항 별 코드)

```
10 // 이동하려는 곳이 유효한 좌표인지 체크하는 함수
11 bool checkXY(int user_x, int mapX, int user_y, int mapY) {
12     bool checkFlag = false;
13     if (user_x >= 0 && user_x < mapX && user_y >= 0 && user_y < mapY) {
14         checkFlag = true;
15     }
16     return checkFlag;
17 }
```

#### 1. 함수 스크린샷 (상)

#### 2. 입력 (블록/함수에 입력되는 변수, 값들과 설명)

- int mapX = 지도의 x 좌표 설정 값
- int mapY = 지도의 y 좌표 설정 값
- int user\_x = 유저 x 값
- int user\_y = 유저 y 값
- bool checkFlag = 유저의 좌표가 설정 좌표 안에 있는지(유효한지) 여부

#### 3. 반환값 (함수의 경우 작성)

- checkFlag (true or false)

4. 결과 (블록/함수가 종료된 결과)

- 유저 좌표의 유효성을 반환한다.

5. 설명 (코드 내 작동 순서, 내용 등 추가 설명)

- 유저의 좌표와 설정 좌표를 비교해 유저의 좌표가 유효한지 확인 후 그 값을 반환한다.

```
19 // 지도와 사용자 위치 출력하는 함수
20 void displayMap(int map[mapY][mapX], int user_x, int user_y) {
21     for (int i = 0; i < mapY; i++) {
22         for (int j = 0; j < mapX; j++) {
23             if (i == user_y && j == user_x) {
24                 cout << " USER |"; // 양 옆 1칸 공백
25             }
26             else {
27                 int posState = map[i][j];
28                 switch (posState) {
29                     case 0:
30                         cout << "      |"; // 6칸 공백
31                         break;
32                     case 1:
33                         cout << "아이템|";
34                         break;
35                     case 2:
36                         cout << " 적  |"; // 양 옆 2칸 공백
37                         break;
38                     case 3:
39                         cout << " 포션 |"; // 양 옆 1칸 공백
40                         break;
41                     case 4:
42                         cout << "목적지|";
43                         break;
44                 }
45             }
46         }
47         cout << endl;
48         cout << " ----- " << endl;
49     }
50 }
```

1. 함수 스크린샷 (상)

2. 입력 (블록/함수에 입력되는 변수, 값들과 설명)

- int map[][] = 전체 지도
- int mapX = 지도의 x 좌표 설정 값
- int mapY = 지도의 y 좌표 설정 값
- int user\_x = 유저 x 값
- int user\_y = 유저 y 값
- int posState = 현재 좌표에 어떤 게 있는지 확인하기 위한 변수

3. 반환값 (함수의 경우 작성)

- 없음

4. 결과 (블록/함수가 종료된 결과)

- 사용자의 위치와 전체 지도를 출력한다.

5. 설명 (코드 내 작동 순서, 내용 등 추가 설명)

- 사용자의 위치, 아이템, 적 등 2차원 배열의 지도를 출력한다.

```
52 // 유저의 위치가 목적지인지 체크하는 함수
53 bool checkGoal(int map[mapY][mapX], int user_x, int user_y) {
54     // 목적지 도착하면
55     if (map[user_y][user_x] == 4) {
56         return true;
57     }
58     return false;
59 }
```

1. 함수 스크린샷 (상)

2. 입력 (블록/함수에 입력되는 변수, 값들과 설명)

- int map[][] = 전체 지도
- int mapX = 지도의 x 좌표 설정 값
- int mapY = 지도의 y 좌표 설정 값
- int user\_x = 유저 x 값
- int user\_y = 유저 y 값

3. 반환값 (함수의 경우 작성)

- 목적지 도착 여부 (true or false)

#### 4. 결과 (블록/함수가 종료된 결과)

- 사용자가 목적지에 도착했는지의 여부를 반환한다.

#### 5. 설명 (코드 내 작동 순서, 내용 등 추가 설명)

- 사용자의 위치가 목적지(4)와 같은 지 확인후 같으면 그 결과를 반환한다.

```

61 // 아이템, 적 포션을 만났을 때의 그에 대한 기능을 동작 및 메시지출력 함수
62 void checkState(int map[mapY][mapX], int& user_x, int& user_y, int& hp) {
63     int position = map[user_y][user_x];
64     switch (position) {
65     case 1: // 아이템을 만났을 때
66         {cout << "아이템이 있습니다. 포션의 위치로 이동하겠습니까? (1: 네 / 그 외: 아니오)" << endl;
67         // 포션 위치로 이동
68         string answer = "";
69         cin >> answer;
70         if (answer == "1"){
71             for (int i = 0; i < mapY; i++) {
72                 for (int j = 0; j < mapX; j++) {
73                     if (map[i][j] == 3) { // 포션 위치 찾기
74                         user_x = j;
75                         user_y = i;
76                         displayMap(map, user_x, user_y);
77                         cout << "'네'를 선택하여 포션이 있는 곳으로 이동했습니다." << endl;
78                         checkState(map, user_x, user_y, hp);
79                         return;
80                     }
81                 }
82             }
83         }
84         else {
85             cout << "'아니오'를 선택했습니다." << endl;
86             return;
87         }
88         break;}
89     case 2: // 적을 만났을 때
90         cout << "적이 있습니다. HP가 2 줄어듭니다." << endl;
91         hp -= 2;
92         break;
93     case 3: // 포션을 만났을 때
94         cout << "포션이 있습니다. HP가 2 늘어납니다." << endl;
95         hp += 2;
96         break;
97     default:
98         break;
99     }
100 }
101

```

#### 1. 함수 스크린샷 (상)

#### 2. 입력 (블록/함수에 입력되는 변수, 값들과 설명)

- int map[][] = 전체 지도
- int mapX = 지도의 x 좌표 설정 값

- int mapY = 지도의 y 좌표 설정 값
- int user\_x = 유저 x 값
- int user\_y = 유저 y 값
- int hp = 사용자의 기본 체력
- int position = 사용자의 위치에 무엇이 있는지 확인하기 위한 변수

3. 반환값 (함수의 경우 작성)

- 없음

4. 결과 (블록/함수가 종료된 결과)

- 사용자가 아이템, 적, 포션을 만났을 때 그에 맞는 메시지 출력과 체력 조정

5. 설명 (코드 내 작동 순서, 내용 등 추가 설명)

- 사용자자의 위치 확인 후 아이템일 경우 포션의 위치로 이동할 것인지 묻고 대답에 따라 이동시킨 후 바뀐 위치와 현재 상태를 출력한다.
- 적/포션 을 만났을 경우 그에 따른 메시지와 hp를 조정한다.

```

103 // 메인 함수
104 int main() {
105     int hp = userHP; // 유저의 초기 체력
106
107     // 0은 빈 공간, 1은 아이템, 2는 적, 3은 포션, 4는 목적지
108     int map[mapY][mapX] = { {0, 1, 2, 0, 4},
109                             {1, 0, 0, 2, 0},
110                             {0, 0, 0, 0, 0},
111                             {0, 2, 3, 0, 0},
112                             {3, 0, 0, 0, 2} };
113
114     // 유저의 위치를 저장할 변수
115     int user_x = 0; // 가로 번호
116     int user_y = 0; // 세로 번호
117
118     // 게임 시작
119     while (true) { // 사용자에게 계속 입력받기 위한 무한 루프
120
121         //사용자가 입력을 저장할 변수
122         string user_input = "";
123
124         cout << "현재 HP: " << hp << endl; // 현재 체력 표시
125         cout << "명령어를 입력하세요 (up, down, left, right, map, finish): ";
126         cin >> user_input;
127
128         if (user_input == "up") {
129             // 위로 한 칸 올라가기
130             user_y -= 1;
131             bool inMap = checkXY(user_x, mapX, user_y, mapY);
132             if (inMap == false) {
133                 cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
134                 user_y += 1;
135             }
136             else {
137                 cout << "위로 한 칸 올라갑니다." << endl;
138                 hp -= 1;
139                 displayMap(map, user_x, user_y);

```

```

140     checkState(map, user_x, user_y, hp);
141 }
142 }
143 else if (user_input == "down") {
144     // 아래로 한 칸 내려가기
145     user_y += 1;
146     bool inMap = checkXY(user_x, mapX, user_y, mapY);
147     if (inMap == false) {
148         cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
149         user_y -= 1;
150     }
151     else {
152         cout << "아래로 한 칸 내려갑니다." << endl;
153         hp -= 1;
154         displayMap(map, user_x, user_y);
155         checkState(map, user_x, user_y, hp);
156     }
157 }
158 else if (user_input == "left") {
159     // 왼쪽으로 이동하기
160     user_x -= 1;
161     bool inMap = checkXY(user_x, mapX, user_y, mapY);
162     if (inMap == false) {
163         cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
164         user_x += 1;
165     }
166     else {
167         cout << "왼쪽으로 이동합니다." << endl;
168         hp -= 1;
169         displayMap(map, user_x, user_y);
170         checkState(map, user_x, user_y, hp);
171     }
172 }
173 else if (user_input == "right") {
174     user_x += 1;
175     bool inMap = checkXY(user_x, mapX, user_y, mapY);
176     if (inMap == false) {
177         cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
178         user_x -= 1;
179     }
180     else {
181         cout << "오른쪽으로 이동합니다." << endl;
182         hp -= 1;
183         displayMap(map, user_x, user_y);
184         checkState(map, user_x, user_y, hp);
185     }
186 }

```



1. 함수 스크린샷 (상)

2. 입력 (블록/함수에 입력되는 변수, 값들과 설명)

- int map[][] = 전체 지도
- int mapX = 지도의 x 좌표 설정 값(상수로 설정)
- int mapY = 지도의 y 좌표 설정 값(상수로 설정)
- int user\_x = 유저 x 값
- int user\_y = 유저 y 값
- int hp = 사용자의 체력
- int userHP = 사용자의 초기 체력 20
- int user\_input = 사용자가 입력한 명령어 저장을 위한 변수

3. 반환값 (함수의 경우 작성)

- 없음

4. 결과 (블록/함수가 종료된 결과)

- 상하좌우로 이동할 좌표가 유효한지 확인 후 그에 따른 문구를 출력하고 유효하다면 이동, 체력을 조정, 지도 출력, 현재상태 출력.

5. 설명 (코드 내 작동 순서, 내용 등 추가 설명)

- 사용자의 hp 설정, 이동 만들기
- 사용자가 입력한 명령어를 확인해 유효성을 확인 후 그에 따른 문구를 출력한다. 유효하다면 이동 후 그에 따른 문구와 hp 조정, 지도와 현재 상태를 출력한다.

```
187     else if (user_input == "map") {
188         //지도 보여주기 함수 호출
189         displayMap(map, user_x, user_y);
190     }
191
192     else if (user_input == "finish") {
193         cout << "종료합니다." << endl;
194         break;
195     }
196     else {
197         cout << "잘못된 입력입니다." << endl;
198         continue;
199     }
```

1. 함수 스크린샷 (상)

2. 입력 (블록/함수에 입력되는 변수, 값들과 설명)

- int user\_x = 유저 x 값
- int user\_y = 유저 y 값
- int usesr\_input = 사용자가 입력한 명령어 저장을 위한 변수

3. 반환값 (함수의 경우 작성)

- 없음

4. 결과 (블록/함수가 종료된 결과)

- 사용자가 입력한 명령어(지도, 종료 혹은 그 외 것들을) 확인하고 그에 따라 지도를 출력하거나 종료메시지 출력 후 프로그램 종료하거나 명령어를 다시 입력받는다.

```
200 // 목적지에 도달했는지 체크
201 // 목적지에 도착하면 hp가 0이어도 성공으로 간주
202 if (checkGoal(map, user_x, user_y)) {
203     cout << "목적지에 도착했습니다! 축하합니다!" << endl;
204     cout << "게임을 종료합니다." << endl;
205     break;
206 }
```

1. 함수 스크린샷 (상)

2. 입력 (블록/함수에 입력되는 변수, 값들과 설명)

- int user\_x = 유저 x 값
- int user\_y = 유저 y 값

3. 반환값 (함수의 경우 작성)

- 없음

4. 결과 (블록/함수가 종료된 결과)

- 목적지에 도착했는지 확인 후 그에 따른 메시지 출력과 프로그램 종료

5. 설명 (코드 내 작동 순서, 내용 등 추가 설명)

- 사용자의 hp 가 0이어도 목적지에 도착했으면 성공이다.

```

208      // HP가 0인지 확인 후 게임 종료
209      if (hp <= 0) {
210          cout << "HP가 0 이하가 되었습니다. 실패했습니다." << endl;
211          break;
212      }
213  }

```

1. 함수 스크린샷 (상)
2. 입력 (블록/함수에 입력되는 변수, 값들과 설명)
  - int hp = 사용자의 체력
3. 반환값 (함수의 경우 작성)
  - 없음
4. 결과 (블록/함수가 종료된 결과)
  - 사용자의 체력을 확인하고 0이라면 메시지 출력과 프로그램 종료

## 4. 테스트

### 1. 기능 별 테스트 결과: (요구사항 별 스크린샷)

<div>현재 HP: 20 명령어를 입력하세요 (up, down, left, right, map, finish): up</div>	<div>지도 밖으로 나가게 되면 에러 메시지 출력</div>
<div>명령어를 입력하세요 (up, down, left, right, map, finish): left 왼쪽으로 이동합니다.  아이템  적    목적지  ----- 아이템      적     -----           ----- USER   적   포션       ----- 포션         적   ----- 현재 HP: 17 명령어를 입력하세요 (up, down, left, right, map, finish): up</div>	<div>이동 명령어(상하좌우)</div>

<pre> 현재 HP: 21 명령어를 입력하세요 (up, down, left, right, map, finish): map   USER   적     목적지   아이템       적     -----   적   포션       ----- 포션         적   ----- 현재 HP: 21 명령어를 입력하세요 (up, down, left, right, map, finish): </pre>	지도 명령어
<pre> 현재 HP: 20 명령어를 입력하세요 (up, down, left, right, map, finish): finish 종료합니다. </pre>	종료 명령어
<pre> 현재 HP: 20 명령어를 입력하세요 (up, down, left, right, map, finish): right 잘못된 입력입니다. </pre>	그 외 명령어
<pre> 명령어를 입력하세요 (up, down, left, right, map, finish): right 오른쪽으로 이동합니다.   USER   적     목적지   아이템       적     -----           -----   적   포션       ----- 포션         적   ----- 아이템이 있습니다. 포션의 위치로 이동하겠습니까? (1: 네 / 그 외: 아니오) 1  아이템  적     목적지  ----- 아이템       적     -----           -----   적   USER       ----- 포션         적   ----- '네'를 선택하여 포션이 있는 곳으로 이동했습니다. 포션이 있습니다. HP가 2 늘어납니다. 현재 HP: 21 </pre>	아이템을 만났을 때 (이동O)
<pre> 현재 HP: 16 명령어를 입력하세요 (up, down, left, right, map, finish): up 위로 한 칸 올라갑니다.  아이템  적     목적지  ----- USER       적     -----           -----   적   포션       ----- 포션         적   ----- 아이템이 있습니다. 포션의 위치로 이동하겠습니까? (1: 네 / 그 외: 아니오) 2 '아니오'를 선택했습니다. </pre>	아이템을 만났을 때 (이동X)

<p>명령어를 입력하세요 (up, down, left, right, map, finish): right 오른쪽으로 이동합니다.</p> <pre>  아이템  적    목적지  ----- 아이템          USER       -----                       -----       적   포션           ----- 포션                       ----- </pre> <p>적이 있습니다. HP가 2 줄어듭니다. 현재 HP: 12</p>	<p>적을 만났을 때</p>
<p>현재 HP: 14 명령어를 입력하세요 (up, down, left, right, map, finish): down 아래로 한 칸 내려갑니다.</p> <pre>  아이템  적    목적지  ----- 아이템           적            -----                       -----       적   USER           ----- 포션                       ----- </pre> <p>포션이 있습니다. HP가 2 늘어납니다. 현재 HP: 15</p>	<p>포션을 만났을 때</p>
<p>현재 HP: 15 명령어를 입력하세요 (up, down, left, right, map, finish): right 오른쪽으로 이동합니다.</p> <pre>  아이템  적     USER   ----- 아이템           적        -----                       -----       적   포션           ----- 포션                       ----- </pre> <p>목적지에 도착했습니다! 축하합니다! 게임을 종료합니다.</p>	<p>목적지에 도착했을 때</p>
<p>현재 HP: 1 명령어를 입력하세요 (up, down, left, right, map, finish): right 오른쪽으로 이동합니다.</p> <pre>  아이템  적    목적지  ----- 아이템           적        -----           USER           -----       적   포션           ----- 포션                       ----- </pre> <p>HP가 0 이하가 되었습니다. 실패했습니다.</p>	<p>체력이 0이 되었을 때</p>

## 2. 최종 테스트 스크린샷: (프로그램 전체 동작 스크린샷)

<div>현재 HP: 20 명령어를 입력하세요 (up, down, left, right, map, finish): down 아래로 한 칸 내려갑니다.  아이템  적    목적지  ----- USER       적     -----           -----   적   포션       ----- 포션         적   ----- 아이템이 있습니다. 포션의 위치로 이동하겠습니까? (1: 네 / 그 외: 아니오) 2 '아니오'를 선택했습니다. 현재 HP: 19 명령어를 입력하세요 (up, down, left, right, map, finish): right 오른쪽으로 이동합니다.  아이템  적    목적지  ----- 아이템  USER     적     -----           -----   적   포션       ----- 포션         적   ----- 현재 HP: 18 명령어를 입력하세요 (up, down, left, right, map, finish): up 위로 한 칸 올라갑니다.   USER   적    목적지  ----- 아이템        적     -----           -----   적   포션       -----</div>	
<div>포션         적   ----- 아이템이 있습니다. 포션의 위치로 이동하겠습니까? (1: 네 / 그 외: 아니오) 1  아이템  적    목적지  ----- 아이템        적     -----           -----   적   USER       ----- 포션         적   ----- '네'를 선택하여 포션이 있는 곳으로 이동했습니다. 포션이 있습니다. hp가 2 늘어납니다. 현재 HP: 19 명령어를 입력하세요 (up, down, left, right, map, finish): right 오른쪽으로 이동합니다.  아이템  적    목적지  ----- 아이템        적     -----           -----   적   포션   USER     ----- 포션         적   ----- 현재 HP: 18 명령어를 입력하세요 (up, down, left, right, map, finish): up 위로 한 칸 올라갑니다.  아이템  적    목적지  ----- 아이템        적     -----         USER     -----   적   포션       -----</div>	

```

포션 |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
-----
현재 HP: 17
명령어를 입력하세요 (up, down, left, right, map, finish): up
위로 한 칸 올라갑니다.
|아이템|  적  |      |      |      |      |      |      |      |      |      |      |      |      |
-----
아이템|      |      |      |      |      |      |      |      |      |      |      |      |      |
-----
|      |      |      |      |      |      |      |      |      |      |      |      |      |
-----
|  적  | 포션 |      |      |      |      |      |      |      |      |      |      |      |
-----
포션 |      |      |      |      |      |      |      |      |      |      |      |      |      |
-----
적이 있습니다. HP가 2 줄어듭니다.
현재 HP: 14
명령어를 입력하세요 (up, down, left, right, map, finish): up
위로 한 칸 올라갑니다.
|아이템|  적  |      |      |      |      |      |      |      |      |      |      |      |      |
-----
아이템|      |      |      |      |      |      |      |      |      |      |      |      |      |
-----
|      |      |      |      |      |      |      |      |      |      |      |      |      |
-----
|  적  | 포션 |      |      |      |      |      |      |      |      |      |      |      |
-----
포션 |      |      |      |      |      |      |      |      |      |      |      |      |      |
-----
현재 HP: 13
명령어를 입력하세요 (up, down, left, right, map, finish): up
맵을 벗어났습니다. 다시 돌아옵니다.
현재 HP: 13
명령어를 입력하세요 (up, down, left, right, map, finish): right
오른쪽으로 이동합니다.
|아이템|  적  |      |      |      |      |      |      |      |      |      |      |      |      |
-----
아이템|      |      |      |      |      |      |      |      |      |      |      |      |      |
-----
포션 |      |      |      |      |      |      |      |      |      |      |      |      |      |
-----
목표지에 도착했습니다! 축하합니다!
게임을 종료합니다.
PS C:\CPP2409>

```

**2. 느낀 점:** 베이스 코드를 올려 주셔서 mud game의 작동 방식을 쉽게 이해할 수 있었다. 덕분에 추가 기능을 구현하는데 어려움이 없었다. 상하좌우, 지도 및 종료를 입력 받을 때 제대로 입력했는데도 불구하고 잘못된 명령어라고 나타나서 당황했지만 영어로 수정해서 실행했더니 잘 작동되었다.