

# 보고서 작성(mud game)

214968(황윤규)

## 1.서론

### 1-1 프로젝트 목적 및 배경

현재까지 배운 내용을 토대로 주어진 추가 기능 요구사항에 맞추어 게임을 개선시키고 반복되는 부분을 함수화시켜 코드를 최적화시킵니다.

### 1-2 목표

Mud게임의 기초적인 구현인 플레이어의 이동과 적, 아이템 등을 구현한다.

## 2.요구사항

### 1-1 사용자 요구사항

유저가 상하좌우로만 이동하며 목적지에 도착하는 게임

### 1-2 기능 요구사항

1. 유저의 체력을 출력합니다.
2. 이동할 위치를 입력받습니다.
3. 입력 받은 위치의 유효성을 체크합니다.
4. 입력 받은 위치로 유저를 이동시킵니다.
5. 이벤트(아이템, 적 등)을 판별하고 실행합니다.
6. 유저의 체력을 감소시킵니다.
7. 목적지에 도달했다면 종료합니다.
8. 만일 유저의 체력이 0이라면 종료합니다.

### 1-3 함수 계획

1. 지도와 사용자의 위치를 출력하는 함수를 만듭니다.
2. 유저의 이동이 유효한지 체크하는 함수를 만듭니다.
3. 목적지에 도달했는지 체크하는 함수를 만듭니다.
4. 지도에서 이벤트를 확인하는 함수를 만듭니다.
5. 유저를 이동시키고, 결과를 출력하는 함수를 만듭니다.

### 3. 설계 및 구현

#### 1. 기능별 구현사항

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 const int mapX = 5;
6 const int mapY = 5;
7
8 // 사용자 정의 함수
9 bool checkXY(int user_x, int mapX, int user_y, int mapY);
10 void displayMap(int map[][mapX], int user_x, int user_y);
11 bool checkGoal(int map[][mapX], int user_x, int user_y);
12 int checkItem(int map[][mapX], int user_x, int user_y);
13
14 // 메인 함수
15 int main() {
16     // 0은 빈 공간, 1은 아이템, 2는 적, 3은 포션, 4는 목적지
17     int map[mapY][mapX] = { {0, 1, 2, 0, 4},
18                             {1, 0, 0, 2, 0},
19                             {0, 0, 0, 0, 0},
20                             {0, 2, 3, 0, 0},
21                             {3, 0, 0, 0, 2}};
22
23     // 유저의 hp를 초기에 20으로 설정함
24     int user_HP = 20;
25
26
27     // 유저의 위치를 저장할 변수
28     int user_x = 0; // 가로 번호
29     int user_y = 0; // 세로 번호
30 }
```

#### 2. 변수

mapX, mapY = 맵의 칸의 개수

int map[mapY][mapX] = 기초적인 맵 세팅

user\_HP = 유저의 체력을 나타냄

user\_x, user\_y = 유저의 위치를 저장함

#### 3. 함수

bool checkXY() = 이동하려는 곳이 유효한 좌표인지 체크하는 함수

void displayMap() = 지도와 유저의 위치를 출력하는 함수

checkGoal() = 유저의 위치가 목적지인지 체크하는 함수

checkItem() = 유저의 위치에 각종 이벤트를 확인하는 함수

#### 1. 기능별 구현사항

```

34 while (1) { // 사용자에게 계속 입력받기 위해 무한 루프
35     // 사용자의 체력이 0이 된다면 "실패"를 출력후 무한루프에서 빠져나옵니다.
36     if(user_HP == 0){
37         cout << "실패 " << endl;
38         break;
39     }
40
41     // 사용자의 입력을 저장할 변수.
42     string user_input = "";
43
44     // 플레이어의 체력을 출력함
45     cout << "현재 HP: " << user_HP << " ";
46     cout << "명령어를 입력하세요 (상,하,좌,우,지도,종료): ";
47     cin >> user_input;
48
49     if (user_input == "상") {
50         movePlayer(map, user_x, user_y, 0, -1, mapX, mapY, user_HP); // 위로 이동
51     }
52     else if (user_input == "하") {
53         movePlayer(map, user_x, user_y, 0, 1, mapX, mapY, user_HP); // 아래로 이동
54     }
55     else if (user_input == "좌") {
56         movePlayer(map, user_x, user_y, -1, 0, mapX, mapY, user_HP); // 왼쪽으로 이동
57     }
58     else if (user_input == "우") {
59         movePlayer(map, user_x, user_y, 1, 0, mapX, mapY, user_HP); // 오른쪽으로 이동
60     }
61     else if (user_input == "지도") {
62         // TODO: 지도 보여주기 함수 호출
63         displayMap(map, user_x, user_y);
64     }
65     else if (user_input == "종료") {
66         cout << "종료합니다.";
67         break;
68     }
69     else {
70         cout << "잘못된 입력입니다." << endl;
71         continue;
72     }
73
74     //이벤트에 따른 hp증감을 checkItem함수를 통해 받습니다.
75     user_HP += checkItem(map, user_x, user_y);
76
77     // 목적지에 도달했는지 체크
78     bool finish = checkGoal(map, user_x, user_y);
79     if (finish == true) {
80         cout << "목적지에 도착했습니다! 축하합니다!" << endl;
81         cout << "게임을 종료합니다." << endl;
82         break;
83     }
84 }

```

## 2. 입력

user\_input = 유저의 행동을 입력받습니다.

## 3. while 문

### - 종료 조건

- 1) user\_HP가 0이 되면 “실패”를 출력 후에 종료합니다.
- 2) user\_input에서 입력받은 값이 “종료”라면 “종료합니다”를 출력 후에 종료합니다.
- 3) checkGoal에서 return 값이 true라면 메시지를 출력 후에 종료합니다.

## 4. if문

- 1) user\_input에 “상”, “하”, “좌”, “우” 입력 시에 movePlayer함수에 map과 user\_x, user\_y, 이동할 x좌표, 이동할 y좌표, mapX, mapY, user\_HP를 보내어 유저를 이동시키고, 그 결과를 보여줍니다.
- 2) user\_input에서 “지도”를 입력하면 displayMap함수에 map, user\_x, user\_y를 보내

어 현재 맵의 상황을 보여줍니다.

3) 이 외의 정보가 입력 시에 “잘못된 입력입니다.”를 출력 후에 continue를 작동합니다.

5. 이 외

1) checkItem함수를 통해 이벤트를 확인하고 그에 따른 HP감소를 user\_HP를 통해 받습니다.

```
// 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;
}
```

1. 입력

- 1) user\_x = 현재 유저의 x좌표
- 2) user\_y = 현재 유저의 y좌표
- 3) mapX = 맵의 x칸의 개수
- 4) mapY = 맵의 y칸의 개수

2. 변수

bool checkFlag = false에서 조건 달성 시에 true로 변환

3. 반환값

- 1) if문 조건 달성 시 true를 리턴
- 2) if문 조건 실패 시 false 리턴

4. 결과

- 1) 맵을 벗어나는 조건인 if문 조건을 만족하면 checkFlag를 true를 반환
- 2) if문 조건을 만족하지 못하면 checkFlag를 false로 반환

5. 설명

유저의 x, y 값의 좌표가 0미만이나 mapX, mapY를 벗어나지 않았다면 true를 반환하여 유효한 좌표임을 나타냄

```
// 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

- 1) user\_x = 현재 유저의 x좌표

2) user\_y = 현재 유저의 y좌표

3) map = 현재 맵

## 2. 변수

int item = 현재 유저의 좌표에 있는 이벤트를 받아옴

## 3. 반환값

1) item의 값이 2일시에 -2를 리턴

2) item의 값이 3일시에 2를 리턴

3) 이외에는 0을 리턴

## 4. 결과

1) item이 1일시에 "아이템이 있습니다."를 출력

2) item의 2일시에 메시지를 출력 후에 -2를 리턴

3) item의 값이 3일시에 메시지를 출력 후에 -2를 리턴

## 5. 설명

아이템을 확인 후에 아이템일 경우에는 0을 리턴

적이 있을 시에는 -2를 리턴

포션이 있을 시에는 2를 리턴하고 그 값을 현재 플레이어의 HP인 user\_HP에 더합니다.

```
// 유저를 이동시키고, 그 결과를 출력하는 함수
void movePlayer(int map[][mapX], int &user_x, int &user_y, int dx, int dy, int mapX, int mapY)
{
    // 유저를 grid 이동시킵니다.
    user_x += dx;
    user_y += dy;

    bool inMap = checkXY(user_x, mapX, user_y, mapY);
    if (!inMap) {
        cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
        user_x -= dx;
        user_y -= dy;
    } else {
        // 방향에 따라 메시지를 출력합니다.
        if (dx == -1) cout << "왼쪽으로 이동합니다." << endl;
        else if (dx == 1) cout << "오른쪽으로 이동합니다." << endl;
        else if (dy == -1) cout << "위로 한 칸 올라갑니다." << endl;
        else if (dy == 1) cout << "아래로 한 칸 내려갑니다." << endl;
        user_HP -= 1;
        // 맵을 보여줍니다.
        displayMap(map, user_x, user_y);
    }
}
```

## 1. 입력

1) user\_x = 현재 유저의 x좌표

2) user\_y = 현재 유저의 y좌표

3) map = 현재 맵

4) mapX = 맵의 x간의 개수

5) mapY = 맵의 y간의 개수

6) dx = 이동할 x 좌표

- 7) dy = 이동할 y 좌표
- 8) user\_HP = 현재 유저의 체력

## 2. 반환값

없음

## 3. 결과

- 1) 이동한 유저의 좌표의 유효성을 체크합니다.
- 2) 이동한 유저의 좌표가 유효하다면 메시지를 출력 후에 이동하고 맵을 보여줍니다.
- 3) 이동한 유저의 좌표가 유효하지 않다면 메시지를 출력합니다.

## 4. 설명

- 1) 우선 현재 유저의 좌표인 user\_x와 user\_y에 상, 하, 좌, 우 일 때 각각 다르게 보낸 dx,dy의 값을 더해 이동한 좌표를 만듭니다.
- 2) checkXY의 함수를 통해 유효성을 체크하고 그 반환값을 inMap변수에 담아놓습니다.
- 3) 만일 유효하지 않다면 메시지를 출력하고 이동한 좌표를 다시 원래대로 되돌립니다.
- 4) 유효하다면 이동한 방향에 맞게 메시지를 출력하고 유저의 HP를 1만큼 감소시킨 후에 displayMap함수를 통해 새롭게 업데이트된 맵을 보여줍니다.

## 4. 테스트

- 1) 잘못된 입력시에 에러 메시지 출력
- 2) 지도 밖으로 나가게 되면 에러 메시지 출력

```

C:\Users\Wjzkc\OneDrive\H...
현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): tkd
잘못된 입력입니다.
현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): 상
맵을 벗어났습니다. 다시 돌아갑니다.
현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): |
  
```

### 3) “지도”를 입력하면 전체 지도와 함께 현재 위치를 출력

```

C:\Users\Wjzkc\OneDrive\H...
현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): 지도
USER |아이템| 적 | |목적지|
-----|-----|-----|-----|
아이템|   |   |   |   |
-----|-----|-----|-----|
      |   |   |   |   |
-----|-----|-----|-----|
      | 적 | 포션 |   |   |
-----|-----|-----|-----|
포션 |   |   |   |   |
-----|-----|-----|-----|
현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): |
  
```

### 4) 유저가 우로 이동

현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우  
오른쪽으로 이동합니다.

	USER		적		목적지
--	------	--	---	--	-----

아이템				적	
-----	--	--	--	---	--

--	--	--	--	--	--

		적	포션		
--	--	---	----	--	--

포션					적
----	--	--	--	--	---

--	--	--	--	--	--

아이템이 있습니다.

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

## 5) 지도에서 “아이템”을 만날 시에 그에 해당하는 메시지 출력

현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우  
오른쪽으로 이동합니다.

	USER		적		목적지
--	------	--	---	--	-----

아이템				적	
-----	--	--	--	---	--

--	--	--	--	--	--

		적	포션		
--	--	---	----	--	--

포션					적
----	--	--	--	--	---

--	--	--	--	--	--

아이템이 있습니다.

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

## 6) 유저가 이동시에 HP가 1감소

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

## 7) 유저가 상, 하, 좌, 우 입력시에 이동

현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우  
오른쪽으로 이동합니다.

	USER		적		목적지
--	------	--	---	--	-----

아이템				적	
-----	--	--	--	---	--

--	--	--	--	--	--

		적	포션		
--	--	---	----	--	--

포션					적
----	--	--	--	--	---

--	--	--	--	--	--

아이템이 있습니다.

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

아래로 한 칸 내려갑니다.

	아이템		적		목적지
--	-----	--	---	--	-----

아이템	USER			적	
-----	------	--	--	---	--

--	--	--	--	--	--

		적	포션		
--	--	---	----	--	--

포션					적
----	--	--	--	--	---

--	--	--	--	--	--

		적	포션		
--	--	---	----	--	--

포션					적
----	--	--	--	--	---

--	--	--	--	--	--

아이템이 있습니다.

현재 HP: 17 명령어를 입력하세요 (상,하,좌,우,지도,종료): 상

위로 한 칸 올라갑니다.

USER	아이템		적		목적지
------	-----	--	---	--	-----

아이템				적	
-----	--	--	--	---	--

--	--	--	--	--	--

		적	포션		
--	--	---	----	--	--

포션					적
----	--	--	--	--	---

현재 HP: 16 명령어를 입력하세요 (상,하,좌,우,지도,종료): |

## 8) 지도에서 “적”과 마주칠시에 메시지를 출력하고 HP를 2가 감소합니다.

```

현재 HP: 15 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 이동합니다.
|아이템| USER |      |목적지|
-----
아이템|      |      | 적   |      |
-----
|      |      |      |      |      |
-----
|      | 적   | 포션 |      |      |
-----
포션 |      |      |      | 적   |
-----
적이 있습니다.
HP가 2 줄어듭니다.
현재 HP: 12 명령어를 입력하세요 (상,하,좌,우,지도,종료): |

```

9) 지도에서 “포션”과 마주칠시에 메시지를 출력하고 HP를 2가 증가합니다.

```

HP가 2 줄어듭니다.
현재 HP: 14 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 이동합니다.
|아이템| 적   |      |목적지|
-----
아이템|      |      | 적   |      |
-----
|      |      |      |      |      |
-----
|      | 적   | USER |      |      |
-----
포션 |      |      |      |      |
-----
포션이 있습니다.
HP가 2 증가합니다
현재 HP: 15 명령어를 입력하세요 (상,하,좌,우,지도,종료): |

```

10) “종료” 입력시에 mud게임을 종료합니다.

```

HP가 2 줄어듭니다.
현재 HP: 11 명령어를 입력하세요 (상,하,좌,우,지도,종료): 종료
종료합니다.
-----
Process exited after 805.1 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . . |

```

11) 체력이 0이 될 시에 메시지를 출력하고 프로그램을 종료합니다.

```

현재 HP: 3 명령어를 입력하세요 (상,하,좌,우,지도,종료): 좌
왼쪽으로 이동합니다.
|아이템| 적   |      |목적지|
-----
아이템|      |      | USER |      |
-----
|      |      |      |      |      |
-----
|      |      | 포션 |      |      |
-----
포션 |      |      |      |      |
-----
적이 있습니다.
HP가 2 줄어듭니다.
실패
-----
Process exited after 148.7 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . . |

```

12) 목적지에 도달시에 메시지를 출력하고 종료합니다.



	아이템	적		USER
아이템			적	
	적	포션		
포션				적

```
Process exited after 7.701 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . . |
```

1. 프로젝트 결과 : 기존 mud게임에 추가 기능 요구사항인 유저는 체력 20을 지니고, 사용  
자가 이동할 때마다 사용자 체력이 1씩 감소하고, 명령문 입력 받을 때 마다 HP출력, HP가  
이 되면 “실패”를 출력하고 종료를 추가 기능 사항으로 조정하고, 겹치는 부분인 유저의 이  
부분을 함수화하여 가독성을 높였습니다.

틱택토와 이어서 여러 가지 프로젝트를 하면서 이번에는 함수화에 관련된 부분에 고민을 많이 하였습니다. 상하좌우 움직이면서 겹치는 부분이 많이 있었고, 처음에는 맵을 벗어났을 때를 함수화 하였는데 생각보다 가독성도 떨어져 전체를 함수화 하여 가동성과 유지 보수적인 측면을 보완하였습니다. 나중에 후에 개인 프로젝트에서도 함수화를 적극 이용해야겠다고 생각했습니다.