mud_game project

214771 정철원

1. 서론

1) 프로젝트 목적 및 배경

- 7주차까지 배운 내용에 대한 실습을 위해 진행

2) 목표

- 간단한 Mud 게임 구현

2. 요구사항

1) 사용자 요구사항

- 유저가 상하좌우로 이동하여 아이템, 포션, 적을 거쳐 목적지에 도착하는 게임

2) 기능 계획

① 사용자에게 “상”, ”하”, ”좌”, ”우”, “지도”, “종료” 중 하나를 입력 받기
- 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 “성공”을 출력하고 종료

• 추가 기능 요구사항

① 유저는 체력 20을 가지고 게임 시작

② 사용자가 이동할 때 마다 사용자 체력 1씩 감소

③ 처음 명령문을 입력 받을 때 마다 HP 함께 출력

④ HP가 0이 되면 “실패”를 출력하고 종료

⑤ 무기/갑옷, 포션, 적을 만났을 때 그에 대한 메시지를 출력

3) 함수 계획

① 메인 함수: 사용자에게 값을 계속 입력받고, 그에 대한 함수 호출

② 지도와 현재 위치 출력 함수: displayMap()

③ 사용자 위치 체크 함수: checkXY()

④ 목적지에 도착 체크 함수: checkGoal()

• 추가 함수 요구사항

① 유저가 만난 대상에 따라 체력이 증감하는 함수 : void checkState(int map[][mapX], int user_x, int user_y, int &hp)

3. 설계 및 구현

```
const int mapX = 5;
const int mapY = 5;
```

입력

• mapX = 맵의 가로

• mapY = 맵의 세로

설명

• 현재 맵은 5*5 크기로 설정됨

```
// 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} };
```

입력

• map = 게임 맵

결과

• 각각의 값은 해당 위치에 있는 상태(대상)

설명

• 0 : 빈공간

• 1 : 아이템

• 2 : 적

• 3 : 포션

• 4 : 목적지

```
// 유저의 위치를 저장할 변수
int user_x = 0; // 가로 번호
int user_y = 0; // 세로 번호
int hp = 20;    // 유저의 초기 체력
```

입력

- user_x : 유저의 현재 가로 위치
- user_y : 유저의 현재 세로 위치
- hp : 유저의 체력

결과

- 맵 상에서의 유저의 위치를 저장하고 유저의 현재 hp상태를 저장할 수 있음

설명

- user_x, user_y는 유저의 현재 위치를 나타내며, 시작 위치는 (0,0)이다.
- hp는 유저의 체력을 나타내며, 초기 값은 20이다.

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

cout << "현재 HP: "<< hp << " 명령어를 입력하세요 (up,down,left,right,map,exit): ";
cin >> user_input;

int prev_x = user_x; // 이전 위치 저장
int prev_y = user_y;
```

입력

- user_input : 유저의 위치를 이동시키는 사용자 명령
- prev_x, prev_y : 이동 전 유저의 위치

결과

- 무한 루프를 통해 유저로부터 명령어를 입력받고 이동 전 위치를 prev_x, prev_y에 저장

설명

- 유저는 up, down, left, right, map, exit 명령어를 사용하여 유저의 위치를 이동시킬 수 있음
- 이동 전 위치를 prev_x와 prev_y에 저장하여 맵을 벗어났을 때 복구하는데 사용됨

```
// 이동에 따른 좌표 변경
if (user_input == "up") user_y -= 1;
else if (user_input == "down") user_y += 1;
else if (user_input == "left") user_x -= 1;
else if (user_input == "right") user_x += 1;
else if (user_input == "map"){
    displayMap(map, user_x, user_y);
    continue;
} //지도 보여주기 함수 호출
else if (user_input == "exit") {
    cout << "종료합니다." << endl;
    break;
} else {
    cout << "잘못된 입력입니다." << endl;
    continue;
}
```

입력

- user_input : 유저의 위치를 이동시키는 사용자 명령
- user_x : 유저의 현재 가로 위치
- user_y : 유저의 현재 세로 위치
- displayMap() : 현재 지도 상태를 보여주는 함수

결과

- 유저로부터 명령어를 입력받고 그에 따른 유저의 위치 이동
- 지도 출력
- 게임 종료

설명

- up down left right를 통해 유저를 상하좌우로 한칸 씩 이동시킬 수 있음
- map을 통해 현재 지도상태를 출력할 수 있음
- exit을 통해 게임을 즉시 종료할 수 있음
- 이외의 값이 입력될 때 잘못된 입력임을 출력하고 다시 루프로 돌아감

```
// 맵 유효성 체크 및 원상복구
if (!checkXY(user_x, mapX, user_y, mapY)) {
    cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
    user_x = prev_x;
    user_y = prev_y;
    continue;
}
```

입력

- mapX = 맵의 가로
- mapY = 맵의 세로
- user_x : 유저의 현재 가로 위치
- user_y : 유저의 현재 세로 위치
- prev_x, prev_y : 이동 전 유저의 위치
- checkXY() : 유저가 맵 범위를 벗어났는지 확인하는 함수

결과

- 유효한 범위인지 아닌지 판단

설명

- checkXY 함수를 통해 유저가 맵 범위를 벗어났는지 확인
- 유효한 범위가 아니면 위치를 원래대로 복구한뒤 다시 루프문 복귀

```
// 이동 메시지 출력
if (user_input == "up") cout << "위로 한 칸 올라갑니다." << endl;
else if (user_input == "down") cout << "아래로 한 칸 내려갑니다." << endl;
else if (user_input == "left") cout << "왼쪽으로 이동합니다." << endl;
else if (user_input == "right") cout << "오른쪽으로 이동합니다." << endl;
```

입력

- user_input : 유저의 위치를 이동시키는 사용자 명령

결과

- 이동 결과 메시지 출력

설명

- 유저가 입력한 방향에 따라 이동 결과 메시지를 출력함

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

// 체력 감소 및 상태 체크
hp -= 1; // 이동 시 체력 1 감소
if (hp <= 0) {
    cout << "HP가 0 이하가 되었습니다. 실패했습니다." << endl;
    break;
}
```

입력

- hp : 유저의 체력
- user_x : 유저의 현재 가로 위치
- user_y : 유저의 현재 세로 위치
- map = 게임 맵
- displayMap() : 현재 맵과 유저의 위치를 출력하는 함수
- checkState() : 유저가 만난 오브젝트에 따라 체력을 조정하는 함수

결과

- 이동할 때 마다 체력 1 감소
- 체력이 0 이하가 되면 실패 메시지 출력 후 프로그램 종료
- 현재 맵과 유저의 위치를 출력
- 유저가 만난 오브젝트에 따라 체력의 변화 출력

설명

- 유저가 이동할 때마다 체력이 1 감소하며, 체력이 0 이하가 되면 게임이 종료됨
- displayMap 함수를 통해 현재 맵과 유저의 위치를 출력
- checkState 함수를 통해 유저가 현재 위치에서 만난 오브젝트에 따라 체력을 조정함

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

입력

- finish : checkGoal의 반환값을 저장하는 변수
- user_x : 유저의 현재 가로 위치
- user_y : 유저의 현재 세로 위치
- map = 게임 맵
- checkGoal() : 목적지에 도달했는지 체크하는 함수

결과

- 목적지에 도달할 시 성공 메시지를 출력 후 게임 종료

설명

- checkGoal 함수를 통해 유저가 목적지에 도달했는지 검사
- 도달했다면 성공 메시지를 출력하고 게임을 종료함

```
// 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

- int map[][mapX] : 전체 맵 정보가 저장된 2차원 배열
- user_x : 유저의 현재 가로 위치
- user_y : 유저의 현재 세로 위치
- map : 게임 맵
- posState : 맵의 각 행렬의 값을 저장

반환값

- 없음

결과

- 전체 맵을 출력하고 유저의 위치를 USER로 표시함

설명

- 5*5 맵을 반복문을 통해 출력함
- 각 위치에 따라 “공백”, “아이템”, “적”, “포션”, “목적지” 등을 표시하고,

유저가 현재 위치한 좌표는 “USER”로 표시하여 시각적으로 확인할 수 있게 함.

```
// 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;
}
```

입력

• user_x : 유저의 현재 가로 위치

• user_y : 유저의 현재 세로 위치

• mapX, mapY : 맵의 가로 세로 크기

• checkFlag : 유효범위 조건의 참 거짓 값을 저장함

반환값

• 유효한 좌표면 true, 아니면 false 값을 반환

결과

• 입력된 위치가 맵의 유효한 범위 내에 있는지 검사하여 결과 반환

설명

• 유저의 위치가 맵의 크기를 벗어나는지 검사하여 벗어나지 않는 경우 true, 벗어나는 경우 false를 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

• user_x : 유저의 현재 가로 위치

• user_y : 유저의 현재 세로 위치

• map[][mapX] : 맵 정보가 저장된 2차원 배열

반환값

• 유저가 목적지에 도달했는지 여부를 반환

결과

• 유저가 목적지에 도달했다면 true, 아니면 false를 반환

설명

• 유저의 위치가 4인지 검사하고, 목적지에 도달했는지 여부를 반환함

```
// 유저가 만난 대상에 따라 체력이 증감하는 함수
void checkState(int map[][mapX], int user_x, int user_y, int &hp) {
    int state = map[user_y][user_x];
    switch (state) {
        case 1:
            cout << "아이템이 있습니다." << endl;
            break;
        case 2:
            hp -= 2;
            cout << "적이 있습니다. HP가 2 줄어듭니다." << endl;
            break;
        case 3:
            hp += 2;
            cout << "포션이 있습니다. HP가 2 늘어납니다." << endl;
            break;
        default:
            break;
    }

}
```

입력

• user_x : 유저의 현재 가로 위치

• user_y : 유저의 현재 세로 위치

• map[][mapX] : 맵 정보가 저장된 2차원 배열

• &hp : 유저의 체력 (참조이므로 값이 변할 수 있다)

• state : map[user_y][user_x] 에 저장되어 있는 값을 저장함

반환값

• 없음

결과

• 유저가 만난 오브젝트에 따라 체력이 증감되고, 그에 따라 메시지가 출력됨

설명

• 유저의 위치에 따라 어떤 오브젝트를 만나는지 확인하고, 체력을 조정함

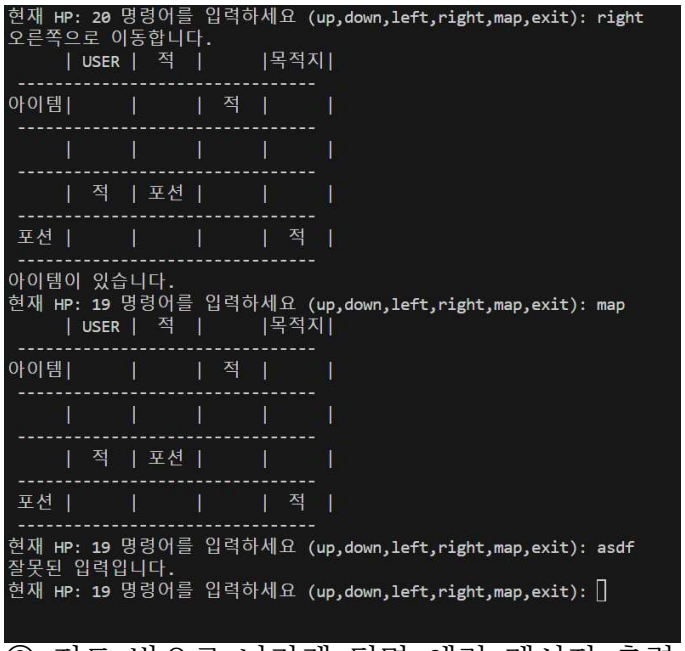
• map[user_y][user_x]값을 확인하여 1이면 아이템을 발견했다는 메시지를 출력하고, 2면 적을 만나 체력이 2 감소, 3이면 포션을 얻어 체력이 2 증가함.

4. 테스트

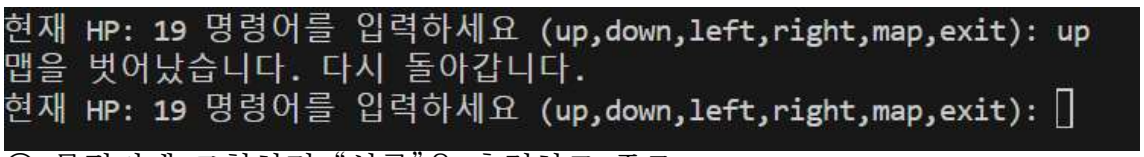
1 기능별 테스트 결과

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

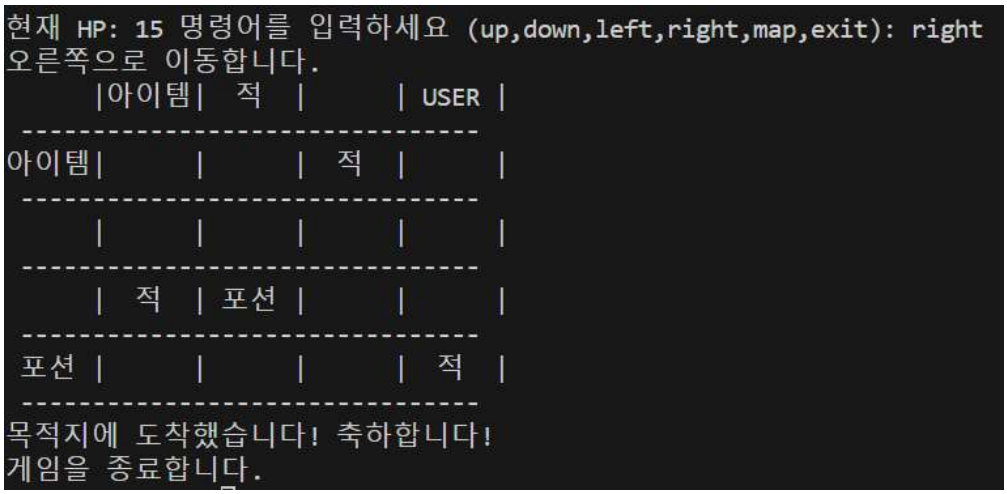
- 상/하/좌/우 입력 시 해당 방향으로 이동 후 지도 출력
- “지도”를 입력하면 전체 지도와 함께 현재 위치를 출력
- 이 중 다른 것을 입력하면 에러 메시지 출력 후 재입력 요청



② 지도 밖으로 나가게 되면 에러 메시지 출력



③ 목적지에 도착하면 “성공”을 출력하고 종료



• 추가 기능 요구사항

① 유저는 체력 20을 가지고 게임 시작

```
현재 HP: 20 명령어를 입력하세요 (up,down,left,right,map,exit):
```

② 사용자가 이동할 때 마다 사용자 체력 1씩 감소

③ 처음 명령문을 입력 받을 때 마다 HP 함께 출력

```
현재 HP: 20 명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다.
     | USER |  적  |      |목적지|
 -------------------------------
아이템|      |      |  적  |      |
 -------------------------------
     |      |      |      |      |
 -------------------------------
     |  적   | 포션 |      |      |
 -------------------------------
 포션 |      |      |      |  적  |
 -------------------------------
아이템이 있습니다.
현재 HP: 19 명령어를 입력하세요 (up,down,left,right,map,exit):
```

④ HP가 0이 되면 “실패”를 출력하고 종료

```
현재 HP: 3 명령어를 입력하세요 (up,down,left,right,map,exit): left
왼쪽으로 이동합니다.
     |아이템| USER |      |목적지|
 -------------------------------
아이템|      |      |  적  |      |
 -------------------------------
     |      |      |      |      |
 -------------------------------
     |  적   | 포션 |      |      |
 -------------------------------
 포션 |      |      |      |  적  |
 -------------------------------
적이 있습니다. HP가 2 줄어듭니다.
HP가 0 이하가 되었습니다. 실패했습니다.
```

⑤ 무기/갑옷, 포션, 적을 만났을 때 그에 대한 메시지를 출력

```
현재 HP: 20 명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다.
     | USER |  적  |      |목적지|
 -------------------------------
아이템|      |      |  적  |      |
 -------------------------------
     |      |      |      |      |
 -------------------------------
     |  적  | 포션 |      |      |
 -------------------------------
 포션 |      |      |      |  적  |
 -------------------------------
아이템이 있습니다.
```

```
현재 HP: 19 명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다.
     |아이템| USER |      |목적지|
 -------------------------------
아이템|      |      |  적  |      |
 -------------------------------
     |      |      |      |      |
 -------------------------------
     |  적  | 포션 |      |      |
 -------------------------------
 포션 |      |      |      |  적  |
 -------------------------------
적이 있습니다. HP가 2 줄어듭니다.
현재 HP: 16 명령어를 입력하세요 (up,down,left,right,map,exit): down
```

```
현재 HP: 14 명령어를 입력하세요 (up,down,left,right,map,exit): down
아래로 한 칸 내려갑니다.
     |아이템|  적  |      |목적지|
 -------------------------------
아이템|      |      |  적  |      |
 -------------------------------
     |      |      |      |      |
 -------------------------------
     |  적  | USER |      |      |
 -------------------------------
 포션 |      |      |      |  적  |
 -------------------------------
포션이 있습니다. HP가 2 늘어납니다.
현재 HP: 15 명령어를 입력하세요 (up,down,left,right,map,exit):
```

3) 함수 계획

① 메인 함수: 사용자에게 값을 계속 입력받고, 그에 대한 함수 호출

② 지도와 현재 위치 출력 함수: displayMap()

```
현재 HP: 15 명령어를 입력하세요 (up,down,left,right,map,exit): map
     |아이템|  적  |      |목적지|
 ------------------------------
아이템|      |      |  적  |      |
 ------------------------------
     |      |      |      |      |
 ------------------------------
     |  적  | USER |      |      |
 ------------------------------
 포션 |      |      |      |  적  |
 ------------------------------
현재 HP: 15 명령어를 입력하세요 (up,down,left,right,map,exit):
```

③ 사용자 위치 체크 함수: checkXY()

```
현재 HP: 15 명령어를 입력하세요 (up,down,left,right,map,exit): down
아래로 한 칸 내려갑니다.
     |아이템|  적  |      |목적지|
 ------------------------------
아이템|      |      |  적  |      |
 ------------------------------
     |      |      |      |      |
 ------------------------------
     |  적  | 포션 |      |      |
 ------------------------------
 포션 |      | USER |      |  적  |
 ------------------------------
현재 HP: 14 명령어를 입력하세요 (up,down,left,right,map,exit): down
맵을 벗어났습니다. 다시 돌아갑니다.
현재 HP: 14 명령어를 입력하세요 (up,down,left,right,map,exit):
```

④ 목적지에 도착 체크 함수: checkGoal()

```
현재 HP: 9 명령어를 입력하세요 (up,down,left,right,map,exit): up
위로 한 칸 올라갑니다.
     |아이템|  적  |      | USER |
 ------------------------------
아이템|      |      |  적  |      |
 ------------------------------
     |      |      |      |      |
 ------------------------------
     |  적  | 포션 |      |      |
 ------------------------------
 포션 |      |      |      |  적  |
 ------------------------------
목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.
```

• 추가 함수 요구사항

① 유저가 만난 대상에 따라 체력이 증감하는 함수 : void checkState(int map[][mapX], int user_x, int user_y, int &hp)

```
현재 HP: 20 명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다.
     | USER |  적  |      |목적지|
 ------------------------------
아이템|      |      |  적  |      |
 ------------------------------
     |      |      |      |      |
 ------------------------------
     |  적  | 포션 |      |      |
 ------------------------------
 포션 |      |      |      |  적  |
 ------------------------------
아이템이 있습니다.
```

```
현재 HP: 19 명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다.
     |아이템| USER |      |목적지|
 ------------------------------
아이템|      |      |  적  |      |
 ------------------------------
     |      |      |      |      |
 ------------------------------
     |  적  | 포션 |      |      |
 ------------------------------
 포션 |      |      |      |  적  |
 ------------------------------
적이 있습니다. HP가 2 줄어듭니다.
현재 HP: 16 명령어를 입력하세요 (up,down,left,right,map,exit): down
```

```
현재 HP: 14 명령어를 입력하세요 (up,down,left,right,map,exit): down
아래로 한 칸 내려갑니다.
     |아이템|  적  |      |목적지|
 ------------------------------
아이템|      |      |  적  |      |
 ------------------------------
     |      |      |      |      |
 ------------------------------
     |  적  | USER |      |      |
 ------------------------------
 포션 |      |      |      |  적  |
 ------------------------------
포션이 있습니다. HP가 2 늘어납니다.
현재 HP: 15 명령어를 입력하세요 (up,down,left,right,map,exit):
```

2 최종 테스트 스크린샷

```
현재 HP: 20 명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다.
      | USER |  적  |      |목적지|
 ------------------------------
아이템|      |      |  적  |      |
 ------------------------------
      |      |      |      |      |
 ------------------------------
      |  적  | 포션 |      |      |
 ------------------------------
 포션 |      |      |      |  적  |
 ------------------------------
아이템이 있습니다.
현재 HP: 19 명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다.
      |아이템| USER |      |목적지|
 ------------------------------
아이템|      |      |  적  |      |
 ------------------------------
      |      |      |      |      |
 ------------------------------
      |  적  | 포션 |      |      |
 ------------------------------
 포션 |      |      |      |  적  |
 ------------------------------
적이 있습니다. HP가 2 줄어듭니다.
현재 HP: 16 명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다.
      |아이템|  적  | USER |목적지|
 ------------------------------
아이템|      |      |  적  |      |
 ------------------------------
      |      |      |      |      |
 ------------------------------
      |  적  | 포션 |      |      |
 ------------------------------
 포션 |      |      |      |  적  |
 ------------------------------
현재 HP: 15 명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다.
      |아이템|  적  |      | USER |
 ------------------------------
아이템|      |      |  적  |      |
 ------------------------------
      |      |      |      |      |
 ------------------------------
      |  적  | 포션 |      |      |
 ------------------------------
 포션 |      |      |      |  적  |
 ------------------------------
목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.
```

5. 결과 및 결론

1) 프로젝트 결과

- mud game을 만들었다.

2) 느낀 점

- 1주차부터 7주차까지 배운 내용을 바탕으로 MUD 게임을 완성했습니다. 첫 번째 프로젝트였던 Tic Tac Toe에서는 단순한 for문과 배열만으로도 혼란을 겪어 프로젝트를 진행하는 데 많은 어려움이 있었습니다. 하지만 그 경험 덕분에 이번 MUD 게임을 만들 때는 보다 더 순조롭게 개발을 이어갈 수 있었습니다. 이 과정에서 for문과 배열에 대한 이해는 물론, 함수의 동작 원리와 순서도 더욱 명확히 익힐 수 있었으며, 참조자(&)를 사용해 주소에 대한 개념 또한 깊이 이해하게 되었습니다. 두 번의 프로젝트 과제를 통해 점점 실력이 향상되는 것을 느끼며, 이 향상된 프로그래밍 실력을 기반으로 완성할 최종 프로젝트의 결과물이 기대됩니다.