Week9 Mud Game 프로젝트 보고서

소프트웨어공학과 214784 이상현

1. 서론

    1. 프로젝트 목적 및 배경: 중간고사 범위까지 학습한 내용의 이해도를 파악하기 위한 실습 진행

    2. 목표: 간단한 Mud Game 구현 및 구체화

2. 요구사항

    1. 사용자 요구사항: 상하좌우로 이동하며 유저를 목적지까지 이동시키기는 게임

    2. 기능 계획:

        ① 사용자에게 "up", "down", "left", "right", "map", "quit" 중 하나를 입력 받기

        • up/down/left/right 입력시 해당 방향으로 이동 후 지도 출력

        • "map"을 입력하면 전체 지도와 함께 현재 위치를 출력

        • "quit"을 입력하면 게임 종료

        • 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

    3. 함수 계획

        ① 메인 함수: 사용자에게 값을 입력 받고, 게임을 진행하거나 종료함

        ② 지도와 현재 위치 출력 함수: displayMap()

        ③ 사용자 위치 체크 함수: checkXY()

        ④ 목적지에 도착 체크 함수: checkGoal()

        ⑤ 아이템, 포션, 적을 만났을 때 상태 변화 함수: checkState()

        ⑥ 사용자 위치 이동 함수: userMove()

    4. 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 아이템, 포션, 적을 만났을 때 그에 대한 메시지를 출력

- 예) {X}가 있습니다.
- 적을 만날 경우 HP가 줄어들고 그에 대한 메시지를 출력

3. 설계 및 구현

1. 기능 별 구현 사항

① 사용자에게 "up", "down", "left", "right", "map", "quit" 중 하나를 입력받기

• up/down/left/right 입력시 해당 방향으로 이동 후 지도 출력

• "map"을 입력하면 전체 지도와 함께 현재 위치를 출력

• "quit"을 입력하면 게임 종료

• 이 중 다른 것을 입력하면 에러 메시지 출력 후 재입력 요청

1. 코드 블록 스크린샷

```
// 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
    int hp = 20; // 유저의 체력

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

        // 사용자의 입력을 저장할 변수
        string user_input = "";
        if (hp <= 0) {
            cout << "HP가 0 이하가 되었습니다. 실패했습니다." << endl;
            cout << "게임을 종료합니다." << endl;
            break;
        }

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

        if (user_input == "quit") {
            cout << "종료합니다.";
            break;
        }

        bool input_check = userMove(map, user_input, user_x, user_y, hp);

        if (input_check == false) {
            continue;
        }
```

```cpp
// 유저의 입력에 따른 위치를 이동시키는 함수
bool userMove(int map[][mapX], string user_input, int& user_x, int& user_y, int& hp) {
    if(user_input == "up") {
        // 입력받은 값에 따라 좌표를 옮기고, hp 1 감소
        user_y -= 1;
        hp -= 1;
        bool inMap = checkXY(user_x, mapX, user_y, mapY);

        // 유효한 이동이 아니라면 원래 좌표로 복귀
        if (inMap == false) {
            cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
            user_y += 1;
            hp += 1;
        }
        // 유효한 이동이라면 이동 성공했다는 문구 출력과 현재 map 상태 출력
        else {
            cout << "위로 한 칸 올라갑니다." << endl;
            displayMap(map, user_x, user_y);
        }
    }
    else if(user_input == "down") {
        user_y += 1;
        hp -= 1;
        bool inMap = checkXY(user_x, mapX, user_y, mapY);

        if (inMap == false) {
            cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
            user_y -= 1;
            hp += 1;
        }
        else {
            cout << "위로 한 칸 내려갑니다." << endl;
            displayMap(map, user_x, user_y);
        }
    }
    else if(user_input == "left") {
        user_x -= 1;
        hp -= 1;
        bool inMap = checkXY(user_x, mapX, user_y, mapY);

        if (inMap == false) {
            cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
            user_x += 1;
            hp += 1;
        }
        else {
            cout << "왼쪽으로 이동합니다." << endl;
            displayMap(map, user_x, user_y);
        }
    }
```

```cpp
    else if(user_input == "right") {
        user_x += 1;
        hp -= 1;
        bool inMap = checkXY(user_x, mapX, user_y, mapY);
        if (inMap == false) {
            cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
            user_x -= 1;
            hp += 1;
        }
        else {
            cout << "오른쪽으로 이동합니다." << endl;
            displayMap(map, user_x, user_y);
        }
    }
    else if (user_input == "map") {
        // TODO: 지도 보여주기 함수 호출
        displayMap(map, user_x, user_y);
        }
    else {
        cout << "잘못된 입력입니다." << endl;
        return false;
    }
    return true;
}
```

2. 입력

   int main() 함수의 입력

   • user_input = 유저의 입력 값 (up, down, left, right, map, quit)

   bool userMove() 함수의 입력

   • int map[][mapX] = 전체 지도
   • string user_input = 유저의 입력을 받는 변수
   • int& user_x = 유저의 현재 x 좌표
   • int& user_y = 유저의 현재 y 좌표
   • int& hp = 유저의 체력 상태
3. 반환값
   bool userMove() 함수에서 user_input이 정상적인 값(up, down, left, right, map)이라면 true를 리턴, 그 외의 값이라면 false를 리턴함.
4. 결과
   • user_input이 방향이라면 유저를 이동 후 전체 지도를 출력함.
   • user_input이 map이면 전체 지도를 출력함.
   • user_input이 quit이면 게임을 종료함.
5. 설명
   • main문에서 전체 지도를 뜻하는 2차원 map 배열을 생성하고, 유저의 위치를 저장할 변수 user_x, user_y와 유저의 체력을 뜻하는 hp를 선언함.
   • while(1)로 게임을 계속 반복하면서 user_input에 입력을 받고, 입력이 "quit"이라면 break;를 통해 게임을 종료함. 그리고 userMove() 함수를 실행시키고 input_check로 리턴을 받아옴. 만약 input_check가 false라면 정상적인 값을 사용자가 입력하지 않았다는 의미이므로 continue; 하여 입력을 다시 받음. 사용자가 방향을 입력했다면 그에 따른 좌표를 늘리거나 줄이고, hp를 1 감소시킴. 그리고 바뀐 좌표가 맵 안에 존재하는지 확인하는 checkXY()를 통해 inMap에 true, false를 저장.
   • userMove()를 통해 inMap이 참이라면 맵 안에 존재한다는 의미이니 좌표를 유지하고 전체 지도를 출력, 거짓이라면 맨 밖에 있다는 뜻이니 좌표와 hp를 원상태로 되돌림.
   • user_input이 map이라면 displayMap()으로 전체 지도를 출력
   • 이에 해당하지 않는 입력이라면 잘못된 입력이라 안내하고, false를 리턴함
   • 입력이 정상적인 값이라면 마지막에 true를 리턴함.
   • hp가 0이하가 되면 안내 문구를 출력하고 break;로 게임을 종료함.

② 지도 밖으로 나가게 되면 에러 메시지 출력

1. 코드 블록 스크린샷

```
// 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;
}
```

2. 입력

   bool checkXY() 함수의 입력

   • int mapX = 전체 지도의 가로축 크기

   • int user_x = 유저의 현재 x 좌표

   • int user_y = 유저의 현재 y 좌표

   • int mapY = 전체 지도의 세로축 크기

3. 반환값

   checkXY() 함수의 반환값

   • checkFlag의 값을 리턴함

4. 결과

   • checkXY() 함수에서는, 입력받은 값을 토대로 이동한 유저의 좌표가 유효한 좌표인지 확인하여 유효하다면 checkFlag에 true를 넣어 리턴, 아니라면 false인 checkFlag를 리턴하여 userMove() 함수 안에 있는 inMap에 저장

   • userMove()를 통해 이동한 좌표가 유효한 값이 아니라면 에러 메시지 출력, 좌표와 체력을 원래 상태로 되돌림.

   • 이동한 좌표가 유효한 값이라면 이동한 방향 출력과 전체 지도 출력

5. 설명

   • checkXY() 함수를 통해 좌표가 유효한지 아닌지 판단하는 inMap 변수 값을 저장하고, userMove() 함수를 통해 inMap이 false라면 에러 메시지를 출력하고 바꿨던 좌표와 체력을 원상태로 되돌림. inMap이 true라면 정상 메시지를 출력하고 지도를 출력함.

③ 목적지에 도착하면 "성공"을 출력하고 종료

1. 코드 블록 스크린샷

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

```
// 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;
}
```

2. 입력

   finish = 게임이 끝났는지 확인하는 bool 변수

   bool checkGoal() 함수의 입력

   • int map[][mapX] = 전체 지도

   • int user_x = 유저의 현재 x 좌표

   • int user_y = 유저의 현재 y 좌표

3. 반환값

   • checkGoal() 함수에서 유저가 목적지에 도착했다면 true를 리턴하고, 그렇지 않다면 false를 리턴함.

4. 결과

   finish가 true면 게임을 종료했다고 출력 및 게임을 종료시킴.

   false면 다시 게임 반복.

5. 설명

   checkGoal() 함수에서 map[user_y][user_x] = 4 일 때(유저의 위치가 목적지를 뜻하는 4와 같다면) 목적지에 도착했음을 뜻하므로 true를 리턴하고, 아니라면 false를 리턴함.

   finish 변수에서 이 값을 받아오고, finish가 true라면 목적지에 도착했음을 출력하고 break;를 통해 while문을 탈출하며 게임을 종료함. finish가 false라면 게임을 다시 반복함.

④ 현재 지도를 출력하는 함수 구현

1. 코드 블록 스크린샷

```
// 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;
    }
}
```

2. 입력

   void displayMap() 함수의 입력

   • int map[][mapX] = 전체 지도
   • int user_x = 유저의 현재 x 좌표
   • int user_y = 유저의 현재 y 좌표
   • int hp = 유저의 체력 상태

3. 결과
   • 현재 지도의 상태와 유저의 현재 위치를 출력함

4. 설명
   • 2차원 배열 map의 값을 posState에 넣어주고, switch문을 통해 해당하는 값에 따라 출력을 바꿔가며 지도를 표시함. 만약 map의 i와 j 값이 user의 x, y 좌표와 같다면 지도에 유저를 출력함.

⑤ 유저의 위치에 따른 유저의 hp 변화

1. 코드 블록 스크린샷

```
// 유저의 위치에 존재하는 아이템, 적, 포션에 따른 상태를 체크하여 hp 변화
hp = checkState(map, user_x, user_y, hp);
```

```
// 유저의 위치에 있는 아이템, 적, 포션에 따른 상태를 변화하는 함수
int checkState(int map[][mapX], int user_x, int user_y, int hp) {
    if(map[user_y][user_x] == 1) {
        cout << "아이템이 있습니다." << endl;
        return hp;
    }
    else if(map[user_y][user_x] == 2) {
        cout << "적이 있습니다. HP가 2 줄어듭니다." << endl;
        return hp - 2;
    }
    else if(map[user_y][user_x] == 3) {
        cout << "포션이 있습니다. HP가 2 늘어납니다." << endl;
        return hp + 2;
    }
    return hp;
}
```

2. 입력

   int checkState() 함수의 입력

   • int map[][mapX] = 전체 지도

   • int user_x = 유저의 현재 x 좌표

   • int user_y = 유저의 현재 y 좌표

3. 반환값

   • 유저의 위치에 아이템이 놓여있다면 현재 hp를 반환

   • 유저의 위치에 포션이 놓여있다면 hp + 2를 반환

   • 유저의 위치에 적이 있다면 hp – 2를 반환

   • 그 외의 값이라면 현재 hp를 반환

4. 결과

   유저의 위치에 따라 hp를 변화하거나 상태 메시지 출력

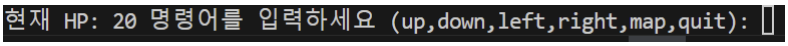
5. 설명

   checkState() 함수를 통해 map[user_y][user_y]의 값이 1 (아이템)이라면 아이템이 있다고 출력하고 현재의 hp를 반환함, 값이 2 (적)이라면 적이 있다고 출력하고 hp를 2 감소시킨 hp-2를 반환함, 값이 3 (포션)이라면 포션이 있다고 출력하고 hp를 2 증가시킨 hp+2를 반환함. 모두 해당되지 않는 경우(비어있는 칸)에 유저가 있다면 현재의 hp를 반환함.

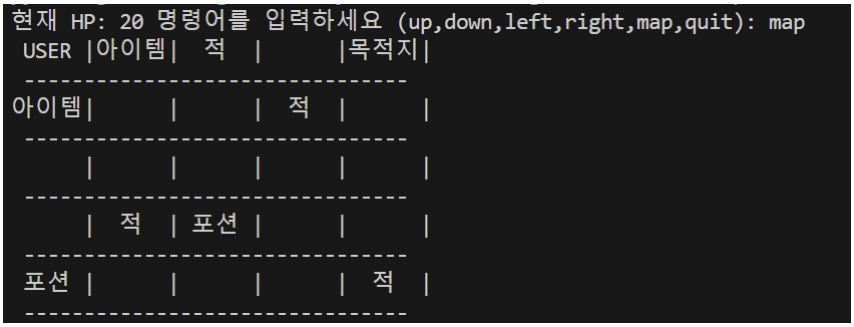
4. 테스트

- (입력에 따라 원하는 결과나 나오는지 확인하는 과정)

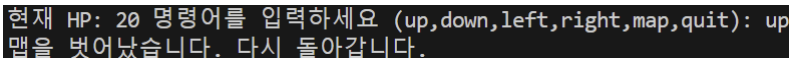
1. 기능 별 테스트 결과: (요구사항 별 스크린샷)
    1. 현재 HP와 사용자의 입력을 받는 문구 출력



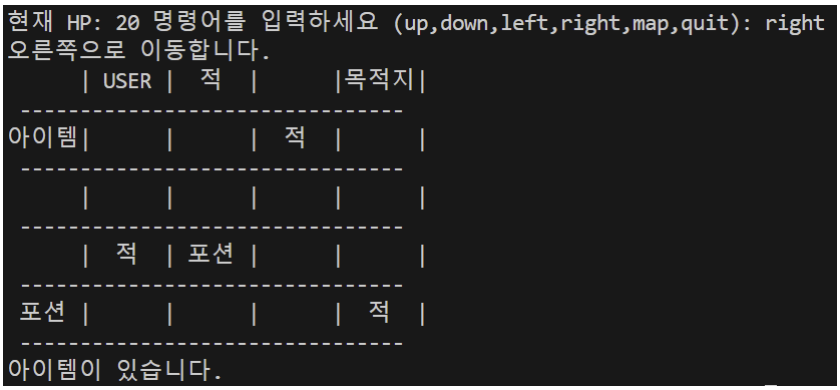
    2. map을 입력했을 때 현재 지도를 출력하기

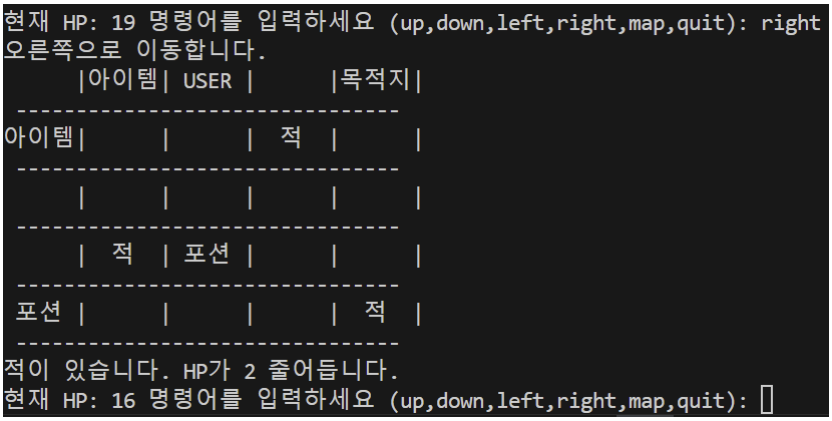


    3. 입력 받은 좌표 유효성 체크



    4. 입력 받은 방향으로 이동 및 사용자의 위치에 있는 요소 출력





```
현재 HP: 14 명령어를 입력하세요 (up,down,left,right,map,quit): down
위로 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,quit):
```

5. 잘못된 입력을 받았을 때 안내 문구 출력

```
현재 HP: 14 명령어를 입력하세요 (up,down,left,right,map,quit): wrong
잘못된 입력입니다.
현재 HP: 14 명령어를 입력하세요 (up,down,left,right,map,quit):
```

6. 주어진 체력 안에 목적지에 도착했을 때 문구 출력 후 종료

```
현재 HP: 13 명령어를 입력하세요 (up,down,left,right,map,quit): up
위로 한 칸 올라갑니다.
     |아이템|  적  |      | USER |
 -------------------------------
아이템|      |      |  적  |      |
 -------------------------------
     |      |      |      |      |
 -------------------------------
     |  적  | 포션 |      |      |
 -------------------------------
 포션 |      |      |      |  적  |
 -------------------------------
목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.
```

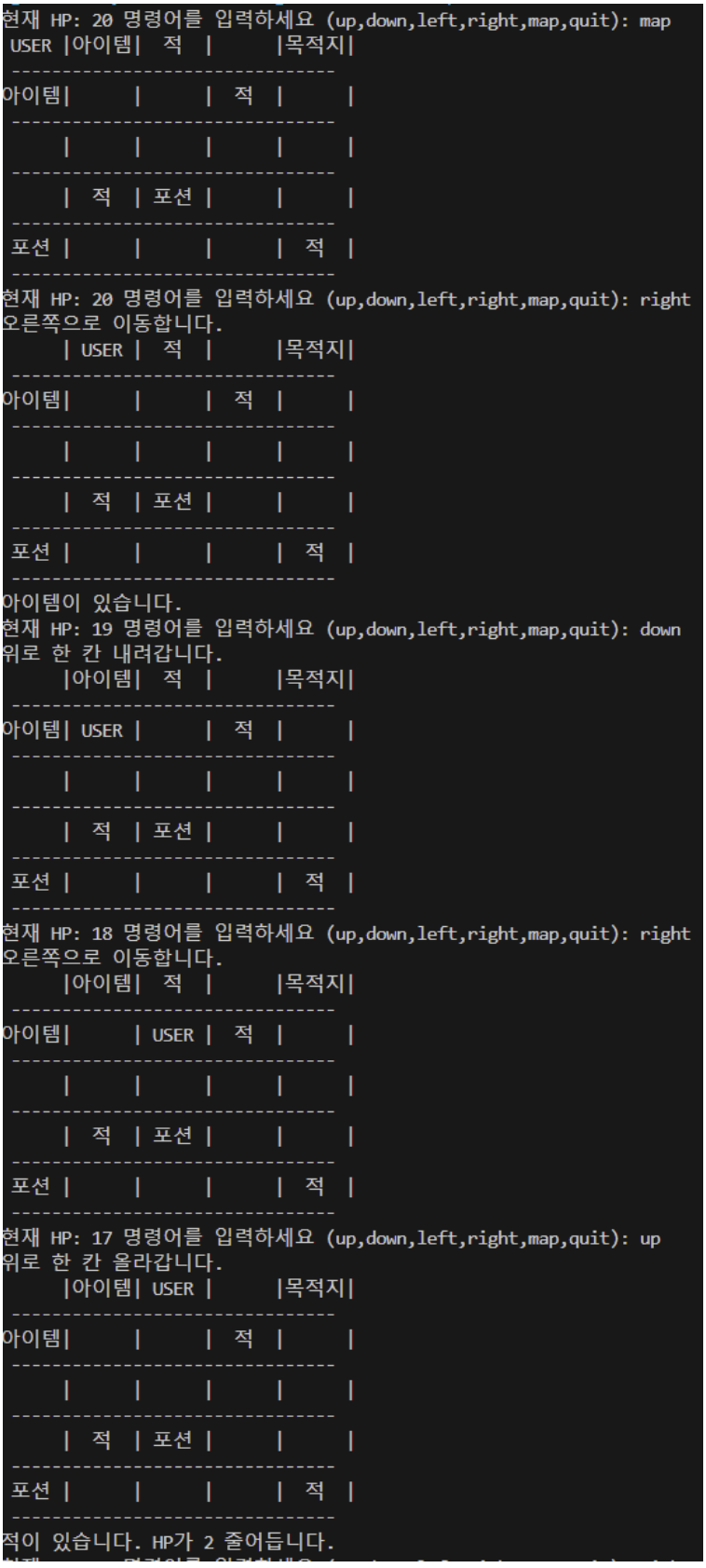
7. 주어진 체력 안에 목적지에 도착하지 못했을 때 문구 출력 후 종료

```
현재 HP: 1 명령어를 입력하세요 (up,down,left,right,map,quit): up
위로 한 칸 올라갑니다.
     |아이템|  적  |      |목적지|
 -------------------------------
아이템|      |      |  적  |      |
 -------------------------------
     |      |      |      | USER |
 -------------------------------
     |  적  | 포션 |      |      |
 -------------------------------
 포션 |      |      |      |  적  |
 -------------------------------
HP가 0 이하가 되었습니다. 실패했습니다.
게임을 종료합니다.
```

8. quit을 입력했을 때 문구 출력 후 게임 종료

```
현재 HP: 17 명령어를 입력하세요 (up,down,left,right,map,quit): quit
종료합니다.
```

2. 최종 테스트 스크린샷: (프로그램 전체 동작 스크린샷)



```
현재 HP: 14 명령어를 입력하세요 (up,down,left,right,map,quit): right
오른쪽으로 이동합니다.
     |아이템|  적   | USER |목적지|
 -------------------------------
아이템|      |      |  적  |      |
 -------------------------------
     |      |      |      |      |
 -------------------------------
     |  적   | 포션 |      |      |
 -------------------------------
 포션 |      |      |      |  적  |
 -------------------------------
현재 HP: 13 명령어를 입력하세요 (up,down,left,right,map,quit): left
왼쪽으로 이동합니다.
     |아이템| USER |      |목적지|
 -------------------------------
아이템|      |      |  적  |      |
 -------------------------------
     |      |      |      |      |
 -------------------------------
     |  적   | 포션 |      |      |
 -------------------------------
 포션 |      |      |      |  적  |
 -------------------------------
적이 있습니다. HP가 2 줄어듭니다.
현재 HP: 10 명령어를 입력하세요 (up,down,left,right,map,quit): right
오른쪽으로 이동합니다.
     |아이템|  적   | USER |목적지|
 -------------------------------
아이템|      |      |  적  |      |
 -------------------------------
     |      |      |      |      |
 -------------------------------
     |  적   | 포션 |      |      |
 -------------------------------
 포션 |      |      |      |  적  |
 -------------------------------
현재 HP: 9 명령어를 입력하세요 (up,down,left,right,map,quit): right
오른쪽으로 이동합니다.
     |아이템|  적   |      | USER |
 -------------------------------
아이템|      |      |  적  |      |
 -------------------------------
     |      |      |      |      |
 -------------------------------
     |  적   | 포션 |      |      |
 -------------------------------
 포션 |      |      |      |  적  |
 -------------------------------
목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.
```