C++프로그래밍및실습

Mud game

머드게임 만들기

제출일자: 2024.11.02
제출자명: 고혜림
제출자학번: 214766

1. 서론

1) 프로젝트 목적 및 배경: 7주차까지 배운 내용에 대한 실습을 진행

2) 목표: 간단한 Mud 게임 구현

2. 요구사항

1) 사용자 요구사항

상하좌우로 이동하며 목적지에 도착하는 게임

2) 기능 계획

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

- "up", "down", "left", "right" 입력 시 해당 방향으로 이동 후 지도 출력
- "map"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 사항

(4) 사용자가 체력 20을 가지고 이동할 때마다 1씩 감소하고 출력

- 0이 되면 게임 종료

(5) 아이템 등을 만났을 때 그에 대한 메시지 출력

3) 함수 계획

(1) 메인 함수: 사용자에게 값을 계속 입력 받고, 그에 대한 함수 호출

(2) 지도와 현재 위치 출력 함수: displayMap()

(3) 사용자 위치 체크 함수: checkXY()

(4) 목적지에 도착 체크 함수: checkGoal()

(5) 아이템 등을 만났을 경우 메시지를 출력하는 함수: checkState()

(6) 사용자의 위치를 변경시키는 함수: moveUser() (계속 반복되는 코드 함수화)

3. 설계 및 구현

1) 기능별 구현 사항

메인 함수

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

1, 코드 스크린샷

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

    // 사용자의 입력을 저장할 변수
    string user_input = "";
    cout << "현재 HP: " << user_hp << " 명령어를 입력하세요 (up,down,left,right,map,quit): ";
    cin >> user_input;
    // cout << "입력된 명령어: " << user_input << endl; // 입력한 명령어가 잘 입력되는지 확인
```

2. 입력

- user_input = 사용자의 입력을 저장한 변수
- user_hp = 사용자의 체력을 저장할 변수 (초기값 = 20)

3. 결과

- 입력하라는 메시지 출력

4. 설명

- 사용자의 입력을 받아 저장
- 사용자의 입력을 받아 저장하는 것을 계속 반복

(1) -1 사용자 입력 시 해당 명령을 수행 후 지도 출력

1. 코드 스크린샷

```
// 이동하는 경우
if (user_input == "up" || user_input == "down" || user_input == "left" || user_input == "right") {
    bool moved = moveUser(map, user_x, user_y, user_hp, user_input);  // 유저 이동
    // 유저가 이동한 경우에만 유저의 위치에 있는것을 확인
    if (moved == true) {
        checkState(map, user_x, user_y, user_hp);  // 유저의 위치 확인
    }
}
// 지도를 보여주는 경우
else if (user_input == "map") {
    // TODO: 지도 보여주기 함수 호출
    displayMap(map, user_x, user_y);
}
// 종료
else if (user_input == "quit") {
    cout << "게임을 종료합니다.";
    break;
}
// 잘못된 입력
else {
    cout << "잘못된 입력입니다." << endl;
    continue;
}
```

2. 입력

- user_input = 사용자의 입력을 저장한 변수
- moved = 사용자가 이동 여부
- moveUser = 사용자를 이동시키고 true를 반환하는 함수
- checkStste = 사용자의 이동이 유효한지 확인하는 함수
- displayMap = 지도를 보여주는 함수

3. 결과

- up, down, right, left 입력 시 해당방향으로 이동
- map 입력 시 지도 출력
- quit 입력 시 종료
- 이 외의 것은 잘못된 입력 출력

4. 설명

- 사용자의 입력이 이동이면 이동을 위한 함수를 호출
- 그 외의 것은 메인함수에서 처리

(2) HP가 0이되는 경우 실패 출력 후 게임 종료

1. 스크린샷

```
// HP가 0이 되었을 경우 실패를 출력하고 게임을 종료
if (user_hp <= 0) {
    cout << "HP가 0이하가 되었습니다. 실패했습니다." << endl;
    cout << "게임을 종료합니다." << endl;
    break;
}
```

2. 입력

- user_input = 사용자의 입력을 저장한 변수

3. 결과

- HP가 0 이하임을 출력
- 게임을 종료

4. 설명

- user_hp가 0 이하인 경우에 메시지를 출력
- break로 반복문을 빠져나와 코드를 종료

(3) 목적지에 도착하면 "성공"을 출력하고 종료

1. 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;
}
```

2. 입력

- map = 전체 지도
- user_x = 사용자의 x값
- user_y = 사용자의 y값
- finish = 목적지에 도달했는지 여부

3. 결과

- 목적지에 도달했다는 메시지 출력
- 게임을 종료

4. 설명

- finish가 true이면 메시지를 출력
- break로 반복문을 빠져나와 게임을 종료

다른 함수들

(4) dispalyMap()

1. 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

- int map[][] = 전체 지도
- user_x = 사용자의 x값
- user_y = 사용자의 y값
- mapX, mapY = 지도의 가로세로 길이

3. 반환값

- 없음

4. 결과

- 전체 지도를 출력
- 사용자의 위치를 출력

5. 설명

- 2차원 배열에 있는 맵을 출력
- 출력하다가 사용자의 위치와 동일한 좌표를 발견할 경우 사용자 정보를 출력

(5) checkXY()

1. 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

- user_x = 사용자의 x값
- mapX = 지도의 가로 길이
- user_y = 사용자의 y값
- mapY = 지도의 세로 길이

3. 반환값

- bool chexkFlag

4. 결과

- 현재 사용자의 자리가 유효한 자리인지 알 수 있음

5. 설명

- 사용자의 자리 값은 (0, 0) ~ (4, 4) 사이여야 함
- 그 사이의 값이라면 checkFlag를 true로 반환

(6) checkGoal()

1. 스크린샷

```
// 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

- int map[][] = 전체 지도
- user_x = 사용자의 x값
- user_y = 사용자의 y값

3. 반환값

- bool (true/false)

4. 결과

- 목적지이면 true 아니면 false

5. 과정

- 지도에서 목적지의 값은 4
- user의 x, y값이 목적지이면 true를 반환

(7) checkState()

1. 스크린샷

```
// 아이템, 적, 포션을 만났을 경우 메세지를 출력하는 함수
void checkState(int map[][mapX], int user_x, int user_y, int& user_hp) {
    switch (map[user_y][user_x]) {
        // 아이템
    case 1:
        cout << "아이템을 만났습니다." << endl;
        break;
        // 적
    case 2:
        cout << "적을 만났습니다." << endl;
        cout << "HP가 2 줄어듭니다." << endl;
        user_hp -= 2;
        break;
        // 포션
    case 3:
        cout << "포션을 얻었습니다." << endl;
        cout << "HP가 2 증가합니다." << endl;
        user_hp += 2;
        break;

    default:
        break;
    }
}
```

2. 입력

- int map[][] = 전체 지도
- user_x = 사용자의 x값
- user_y = 사용자의 y값
- user_hp = 사용자의 체력을 저장할 변수 (초기값 = 20)

3. 반환값

- 없음

4. 결과

- 아이템, 적, 포션을 만날 경우 관련된 메시지를 출력
- 적을 만날 경우 HP를 2 감소
- 포션을 만날 경우 HP를 2 증가

5. 설명

- switch문을 사용하여 1(아이템), 2(적), 3(포션)을 나눠서 출력
- user_hp의 경우 변수를 직접 참조해 함수에서 값을 바꿀 수 있게 함

(8) moveUser()

1. 스크린샷

```
// 유저의 위치를 변경시키는 함수
bool moveUser(int map[][mapX], int& user_x, int& user_y, int& user_hp, string user_input) {
    int temp_x = user_x;
    int temp_y = user_y;

    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;
    }

    if (checkXY(user_x, mapX, user_y, mapY) == false) {
        cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
        user_x = temp_x;
        user_y = temp_y;
        return false;
    } else {
        user_hp -= 1;
        cout << user_input << " 방향으로 이동합니다." << endl;
        displayMap(map, user_x, user_y);
        return true;
    }
}
```

2. 입력

- int map[][] = 전체 지도
- user_x = 사용자의 x값
- user_y = 사용자의 y값
- user_hp = 사용자의 체력을 저장할 변수 (초기값 = 20)
- user_input = 사용자의 입력을 저장한 변수

3. 반환값

- bool (true/false)

4. 결과

- 사용자의 위치 변경
- 전체 지도 출력

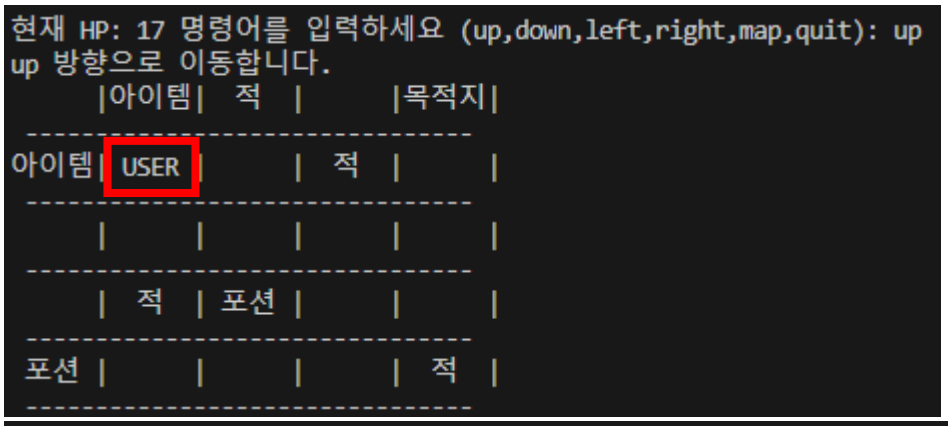
5. 설명

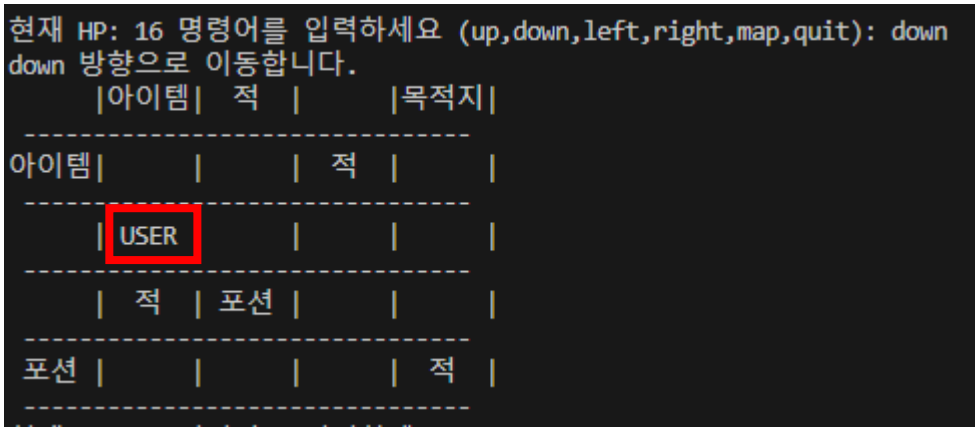
- user_x와 user_y값을 직접 참조해서 입력에 맞게 변경
- checkXY()를 호출해 유효하지 않으면 원래 값으로 돌아감
- 이동이 유효한 경우 hp감소 후 맵 출력

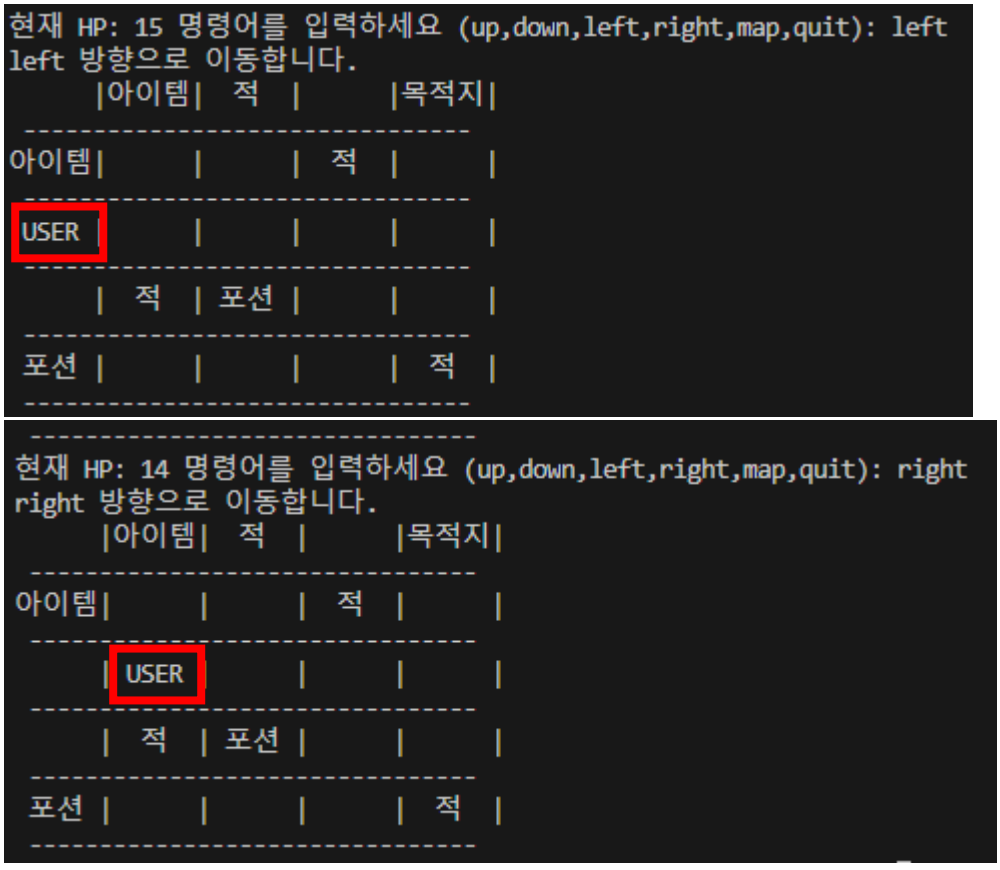
4. 테스트

1) 기능별 테스트 결과

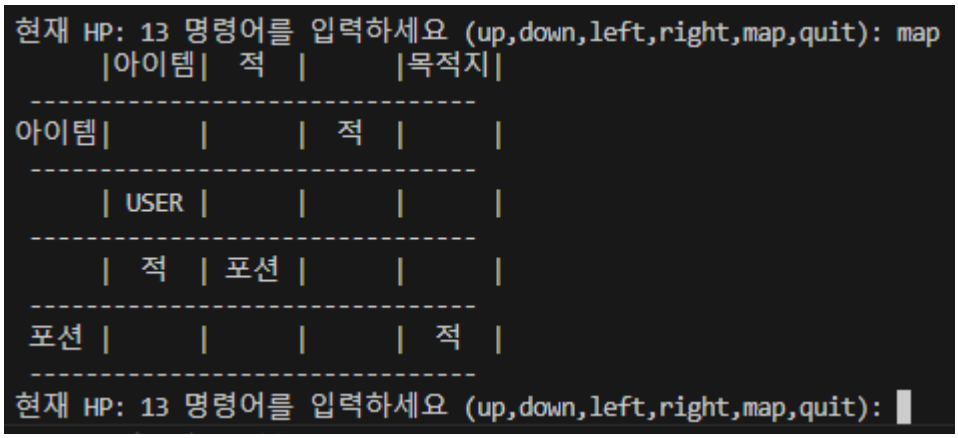
① up/down/left/right 입력 시 해당 방향으로 이동 후 지도 출력



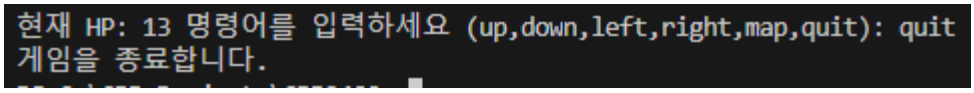




② "map"을 입력하면 전체 지도와 함께 현재 위치를 출력



③ "quit"을 입력하면 게임 종료 메세지와 종료



④ 이 중 다른 것을 입력하면 에러 메시지 출력 후 재 입력 요청

```
현재 HP: 20 명령어를 입력하세요 (up,down,left,right,map,quit): abc
잘못된 입력입니다.
현재 HP: 20 명령어를 입력하세요 (up,down,left,right,map,quit): def
잘못된 입력입니다.
현재 HP: 20 명령어를 입력하세요 (up,down,left,right,map,quit):
```

⑤ 지도 밖으로 나가게 되면 에러 메시지 출력

```
현재 HP: 20 명령어를 입력하세요 (up,down,left,right,map,quit): up
맵을 벗어났습니다. 다시 돌아갑니다.
현재 HP: 20 명령어를 입력하세요 (up,down,left,right,map,quit): left
맵을 벗어났습니다. 다시 돌아갑니다.
현재 HP: 20 명령어를 입력하세요 (up,down,left,right,map,quit):
```

⑥ 목적지에 도착하면 "성공"을 출력하고 종료

```
현재 HP: 15 명령어를 입력하세요 (up,down,left,right,map,quit): right
right 방향으로 이동합니다.
     |아이템|  적  |      | USER |
 ------------------------------
아이템|      |      |  적  |      |
 ------------------------------
     |      |      |      |      |
 ------------------------------
     |  적  | 포션 |      |      |
 ------------------------------
 포션 |      |      |      |  적  |
 ------------------------------
목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.
```

⑦ 사용자가 체력 20을 가지고 이동할 때마다 1씩 감소하고 출력

```
현재 HP: 20 명령어를 입력하세요 (up,down,left,right,map,quit): down
down 방향으로 이동합니다.
     |아이템|  적  |      |목적지|
 ------------------------------
 USER |      |      |  적  |      |
 ------------------------------
     |      |      |      |      |
 ------------------------------
     |  적  | 포션 |      |      |
 ------------------------------
 포션 |      |      |      |  적  |
 ------------------------------
아이템을 만났습니다.
현재 HP: 19 명령어를 입력하세요 (up,down,left,right,map,quit): down
```

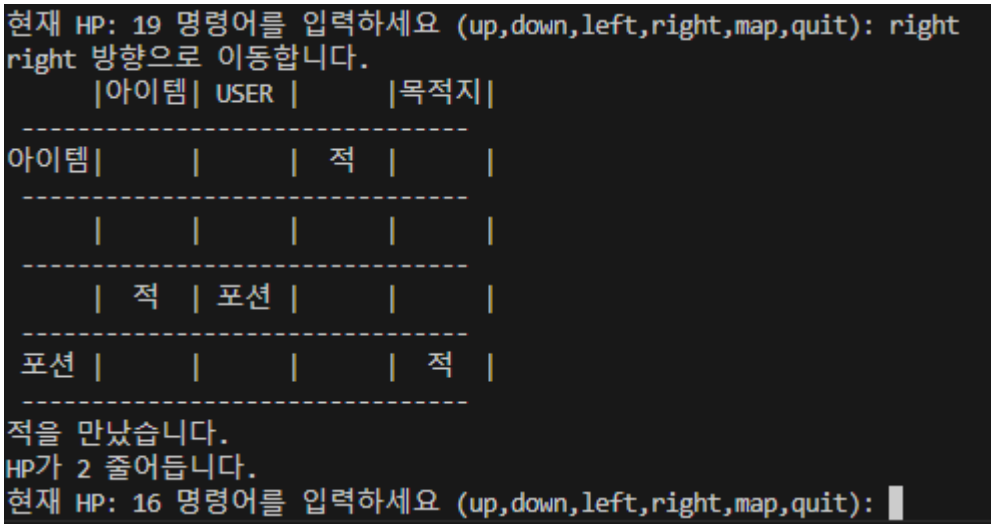
```
현재 HP: 19 명령어를 입력하세요 (up,down,left,right,map,quit): down
down 방향으로 이동합니다.
     |아이템|  적  |      |목적지|
 -------------------------------
아이템|      |      |  적  |      |
 -------------------------------
 USER |      |      |      |      |
 -------------------------------
      |  적  | 포션 |      |      |
 -------------------------------
 포션 |      |      |      |  적  |
 -------------------------------
현재 HP: 18 명령어를 입력하세요 (up,down,left,right,map,quit): down
down 방향으로 이동합니다.
```

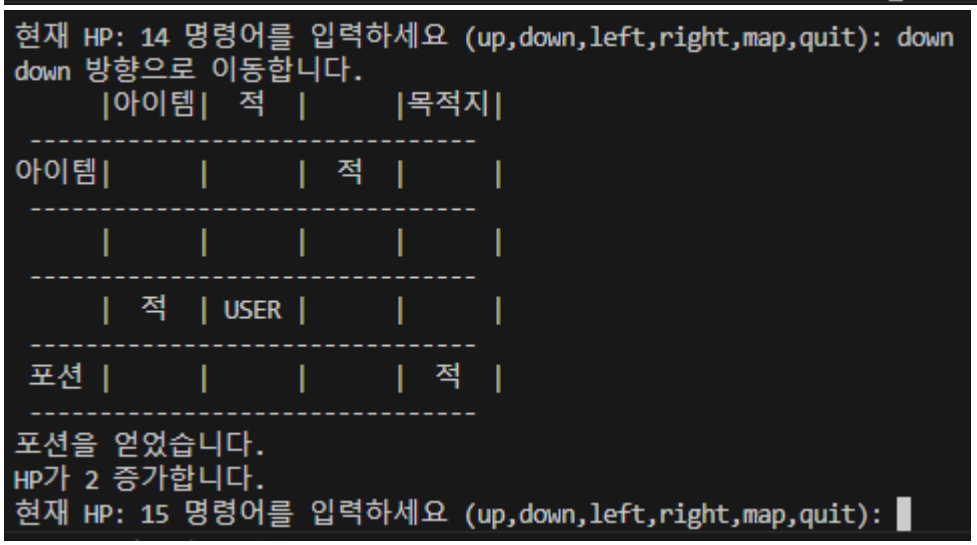
⑧ 체력이 0이 되면 게임 종료

```
현재 HP: 1 명령어를 입력하세요 (up,down,left,right,map,quit): right
right 방향으로 이동합니다.
     |아이템|  적  |      |목적지|
 -------------------------------
아이템|      |      |  적  |      |
 -------------------------------
      |      |      |      |      |
 -------------------------------
      | USER | 포션 |      |      |
 -------------------------------
 포션 |      |      |      |  적  |
 -------------------------------
적을 만났습니다.
HP가 2 줄어듭니다.
HP가 0이하가 되었습니다. 실패했습니다.
게임을 종료합니다.
```

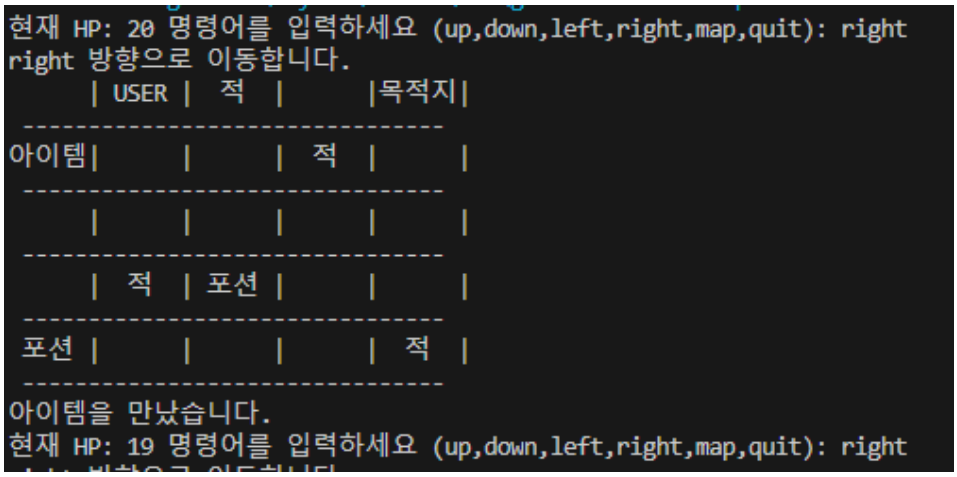
⑨ 아이템, 적, 포션을 만났을 때 그에 대한 메시지 출력

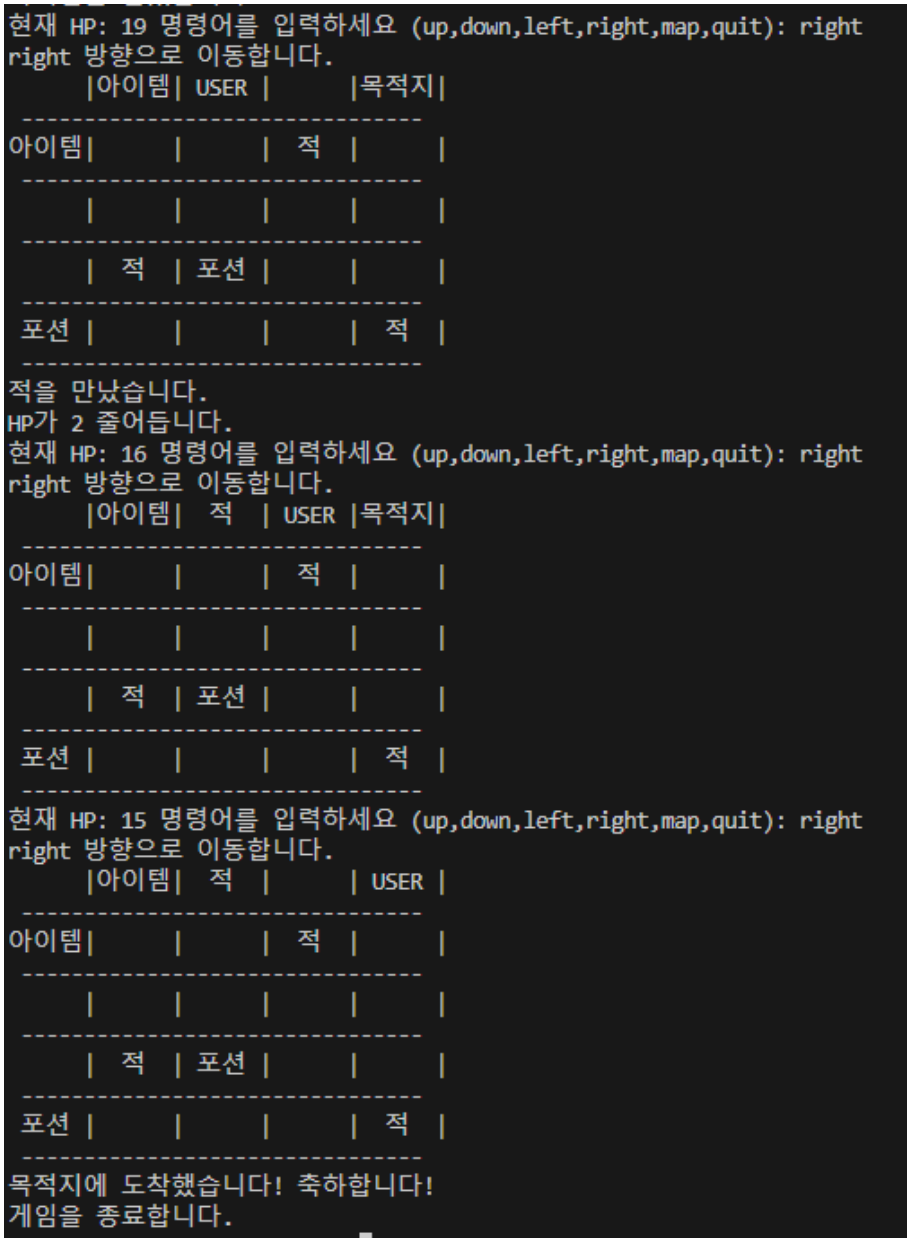
```
현재 HP: 20 명령어를 입력하세요 (up,down,left,right,map,quit): right
right 방향으로 이동합니다.
     | USER |  적  |      |목적지|
 -------------------------------
아이템|      |      |  적  |      |
 -------------------------------
      |      |      |      |      |
 -------------------------------
      |  적  | 포션 |      |      |
 -------------------------------
 포션 |      |      |      |  적  |
 -------------------------------
아이템을 만났습니다.
현재 HP: 19 명령어를 입력하세요 (up,down,left,right,map,quit):
```


현재 HP: 19 명령어를 입력하세요 (up,down,left,right,map,quit): right
right 방향으로 이동합니다.
|아이템| USER | |목적지|
아이템| | | 적 | |
| | | | |
| 적 | 포션 | | |
포션 | | | | 적 |
적을 만났습니다.
HP가 2 줄어듭니다.
현재 HP: 16 명령어를 입력하세요 (up,down,left,right,map,quit):


현재 HP: 14 명령어를 입력하세요 (up,down,left,right,map,quit): down
down 방향으로 이동합니다.
|아이템| 적 | |목적지|
아이템| | | 적 | |
| | | | |
| 적 | USER | | |
포션 | | | | 적 |
포션을 얻었습니다.
HP가 2 증가합니다.
현재 HP: 15 명령어를 입력하세요 (up,down,left,right,map,quit):

2) 최종 테스트 스크린샷


현재 HP: 20 명령어를 입력하세요 (up,down,left,right,map,quit): right
right 방향으로 이동합니다.
| USER | 적 | |목적지|
아이템| | | 적 | |
| | | | |
| 적 | 포션 | | |
포션 | | | | 적 |
아이템을 만났습니다.
현재 HP: 19 명령어를 입력하세요 (up,down,left,right,map,quit): right



5. 결과 및 결론

1) 프로젝트 결과

Mud game에 사용자의 체력을 추가했다. 아이템, 적, 포션을 만날 때 나타나는 이벤트를 추가했다. 이동에서 반복되는 코드를 함수화 했다.

2) 느낀점

한글 입력이 제대로 되지 않아서 걱정했지만 입력을 영어로 바꿔서 해결했다. 내 vscode가 한글 인식을 해줬으면 좋겠다.