

보고서

학과 : 소프트웨어공학과

학번 : 213182

이름 : 박태은

주제 : Mud Game C++

1. 서론

1. 프로젝트 목적 및 배경 : 7 주차까지 배운 내용에 대한 실습을 위해 진행
2. 목표 : 간단한 Mud 게임 구현

2. 요구사항

1. 사용자 요구사항 : 유저가 상하좌우로만 이동하며 목적지에 도착하는 게임
2. 기능 계획
 - 1) 사용자에게 "상", "하", "좌", "우", "지도", "종료"중 하나를 입력 받기
 - 상/하/좌/우 입력 시 해당방향으로 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
 - 4) 유저가 체력 20 가지고 게임 시작
 - 5) 사용자가 이동할 때 마다 사용자 체력 1씩 감소
 - 6) 처음 명령문을 입력 받을 때 마다 HP 함께 출력
 - 7) HP가 0이 되면 "실패"를 출력하고 종료
 - 8) 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()
- 5) 체력 출력 함수: HP_print()
- 6) 체력 변화 함수: HP_change()
- 7) 아이템/적/포션 메시지 출력 함수: checkState()

3. 설계 및 구현

1. 코드블록 스크린샷

```
// 체력 변화 함수
// 좌표값에 따라 switch문으로 case별로 체력떨어트리고 HP값을 return한다.
int HP_change(int HP, int map[][mapX], int user_x, int user_y){

    int posState = map[user_y][user_x];
    switch (posState){
        case 0:
            HP -= 1;
            break;
        case 1:
            HP -= 1;
            break;
        case 2:
            HP -= 3;
            break;
        case 3:
            HP += 1;
            break;
        case 4:
            HP -= 1;
            break;
    }

    return HP;
}
```

2. 입력(블록/함수에 입력되는 변수, 값들 설명)

- HP = 현재 체력
- map[][] = 전체 지도
- user_x = 유저 x값
- user_y = 유저 y값

3. 반환값(함수의 경우 작성)

- HP

4. 결과(블록/함수가 종료된 결과)

- 현재 좌표에 따라 HP의 값에 변화를 줌

5. 설명(코드 내 작동 순서, 내용 등 추가 설명)

- 현재 좌표값을 switch문에 넣어서 0, 1, 2, 3, 4인 경우 각 변화 작성
- 나머지는 -1로 동일, 2인경우 -3, 3인경우 +1한다.

1. 코드블록 스크린샷

```
// 아이템, 적, 포션 만났을 때 메시지 출력 함수
void checkState (int map[][mapX], int user_x , int user_y ){
    int posState = map[user_y][user_x];
    switch (posState){
        case 0:
            break;
        case 1:
            cout << "아이템이 있습니다." << endl;
            break;
        case 2:
            cout << "적이 있습니다. HP가 2 줄어듭니다." << endl;
            break;
        case 3:
            cout << "포션이 있습니다. HP가 2 늘어납니다." << endl;
            break;
        case 4:
            break;
    }
}
```

2. 입력(블록/함수에 입력되는 변수, 값들 설명)

- map[][] = 전체 지도
- user_x = 유저 x값
- user_y = 유저 y값

3. 반환값(함수의 경우 작성)

- 없음

4. 결과(블록/함수가 종료된 결과)

- 아이템, 포션, 적을 만났을 때 그에 대한 메시지를 출력

5. 설명(코드 내 작동 순서, 내용 등 추가 설명)

- 현재 좌표값을 switch문에 넣어서 0, 1, 2, 3, 4인 경우 각 변화 작성
- 좌표값이 0이나 4의 경우는 따로 출력 없다.
- 좌표값이 1이면 아이템, 2는 적, 3은 포션에 대한 문구를 출력한다.

1. 코드블록 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(블록/함수에 입력되는 변수, 값들 설명)

- map[][] = 전체 지도
- user_x = 유저 x값
- user_y = 유저 y값

3. 반환값(함수의 경우 작성)

- 없음

4. 결과(블록/함수가 종료된 결과)

- 전체 지도를 출력
- 사용자 위치를 출력

5. 설명(코드 내 작동 순서, 내용 등 추가 설명)

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

1. 코드블록 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(블록/함수에 입력되는 변수, 값들 설명)

- mapY = 배열의 행 크기
- mapX = 배열의 열 크기
- user_x = 유저 x값
- user_y = 유저 y값

3. 반환값(함수의 경우 작성)

- checkFlag(true or false)

4. 결과(블록/함수가 종료된 결과)

- 좌표가 유효하면 checkFlag값을 true로, 아니면 false로 반환한다.

5. 설명(코드 내 작동 순서, 내용 등 추가 설명)

- user_x와 user_y가 0보다 크거나 같고, mapX와 mapY보다 작을 때 범위가 유효하다.
- 유효한 경우 checkFlag에 true를 넣어서 true값을 반환

1. 코드블록 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(블록/함수에 입력되는 변수, 값들 설명)

- map = 전체지도
- user_x = 유저 x값
- user_y = 유저 y값

3. 반환값(함수의 경우 작성)

- true or false
- 결과(블록/함수가 종료된 결과)
 - 목적지에 도착한 경우 true를 반환한다.
 - 설명(코드 내 작동 순서, 내용 등 추가 설명)
 - 조건문으로 좌표값이 4에 도달한 경우 true를 반환,
 - 아닌 경우 false를 반환

1. 코드블록 스크린샷

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

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

    // HP 출력 함수
    HP_print(HP);
    cout << "명령어를 입력하세요 (up, down, left, right, map, quit): ";
    cin >> user_input;
```

- 입력(블록/함수에 입력되는 변수, 값들 설명)
 - user_input = 사용자 입력값
 - HP = 현재 체력
- 결과(블록/함수가 종료된 결과)
 - 현재 HP 출력
 - 명령어 입력 문구 출력
- 설명(코드 내 작동 순서, 내용 등 추가 설명)
 - while문으로 게임 시작한다.
 - HP_print()함수로 현재 HP 출력
 - 명령어 입력 문구 출력
 - user_input 입력받는다.

1. 코드블록 스크린샷

```
if (user_input == "up") {
    // 위로 한 칸 올라가기
    user_y -= 1;
    bool inMap = checkXY(user_x, mapX, user_y, mapY);
    if (inMap == false) {
        cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
        user_y += 1;
    }
    else {
        cout << "위로 한 칸 올라갑니다." << endl;
        displayMap(map, user_x, user_y);
        // 위로 이동시 HP 변화 함수
        HP=HP_change(HP, map, user_x, user_y);
        // 아이템, 적, 포션 만났을 때 메시지 출력 함수
        checkState (map, user_x , user_y );
    }
}
```

2. 입력(블록/함수에 입력되는 변수, 값들 설명)

- user_input = 사용자 입력값
- HP = 현재 체력
- map[][] = 전체 지도
- user_x = 유저 x값
- user_y = 유저 y값
- mapY = 배열의 행 크기
- mapX = 배열의 열 크기

3. 결과(블록/함수가 종료된 결과)

- 맵을 벗어난 경우 : 기존 위치로 다시 돌아감
- 맵을 벗어나지 않는 경우 :
위로 한칸 올라간다는 문구 출력하고, 현재 지도 출력, HP변환, 아이템, 적, 포션
만난 경우 문구 출력

4. 설명(코드 내 작동 순서, 내용 등 추가 설명)

- user_input값이 up인 경우에 조건문에 들어온다.
- 일단 user_y값을 -1하고 checkXY함수로 범위 벗어나는지 확인한다.
- 벗어나는 경우 다시 좌표값에 +1을 해서 되돌리고 맵을 벗어났다고 출력
- 안벗어나는 경우 위로 한칸 올라간다고 출력 후, displayMap()함수로 현재 지도
및 사용자 위치 출력 후, HP_change()함수로 체력변화, checkState()함수로 아이
템, 적, 포션을 만나면 문구를 출력한다.

1. 코드블록 스크린샷

```
else if (user_input == "down") {
    // TODO: 아래로 한 칸 내려가기
    user_y += 1;
    bool inMap = checkXY(user_x, mapX, user_y, mapY);
    if (inMap == false) {
        cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
        user_y -= 1;
    }
    else {
        cout << "아래로 한 칸 내려갑니다." << endl;
        displayMap(map, user_x, user_y);
        // 아래로 이동시 HP 변화 함수
        HP=HP_change(HP, map, user_x, user_y);
        // 아이템, 적, 포션 만났을 때 메시지 출력 함수
        checkState (map, user_x , user_y );
    }
}
```

2. 입력(블록/함수에 입력되는 변수, 값들 설명)

- user_input = 사용자 입력값
- HP = 현재 체력
- map[][] = 전체 지도
- user_x = 유저 x값
- user_y = 유저 y값
- mapY = 배열의 행 크기
- mapX = 배열의 열 크기

3. 결과(블록/함수가 종료된 결과)

- 맵을 벗어난 경우 : 기존 위치로 다시 돌아감
- 맵을 벗어나지 않는 경우 :
아래로 한칸 내려간다는 문구 출력하고, 현재 지도 출력, HP변환, 아이템, 적, 포션 만난 경우 문구 출력

4. 설명(코드 내 작동 순서, 내용 등 추가 설명)

- user_input값이 down인 경우에 조건문에 들어온다.
- 일단 user_y값을 +1하고 checkXY함수로 범위 벗어나는지 확인한다.
- 벗어나는 경우 다시 좌표값에 -1을 해서 되돌리고 맵을 벗어났다고 출력.
- 안벗어나는 경우 아래로 한칸 내려간다고 출력 후, displayMap()함수로 현재 지도 및 사용자 위치 출력 후, HP_change()함수로 체력변화, checkState()함수로 아이템, 적, 포션을 만나면 문구를 출력한다.

1. 코드블록 스크린샷

```
else if (user_input == "left") {
    // TODO: 왼쪽으로 이동하기
    user_x -= 1;
    bool inMap = checkXY(user_x, mapX, user_y, mapY);

    if (inMap == false) {
        cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
        user_x += 1;
    }
    else {
        cout << "왼쪽으로 이동합니다." << endl;
        displayMap(map, user_x, user_y);
        // 왼쪽으로 이동시 HP 변화 함수
        HP=HP_change(HP, map, user_x, user_y);
        // 아이템, 적, 포션 만났을 때 메시지 출력 함수
        checkState (map, user_x , user_y );
    }
}
```

2. 입력(블록/함수에 입력되는 변수, 값들 설명)

- user_input = 사용자 입력값
- HP = 현재 체력
- map[][] = 전체 지도
- user_x = 유저 x값
- user_y = 유저 y값
- mapY = 배열의 행 크기
- mapX = 배열의 열 크기

3. 결과(블록/함수가 종료된 결과)

- 맵을 벗어난 경우 : 기존 위치로 다시 돌아감
- 맵을 벗어나지 않는 경우 :
왼쪽으로 한칸 이동한다는 문구 출력하고, 현재 지도 출력, HP변환, 아이템, 적,
포션 만난 경우 문구 출력

4. 설명(코드 내 작동 순서, 내용 등 추가 설명)

- user_input값이 left인 경우에 조건문에 들어온다.
- 일단 user_x값을 -1하고 checkXY함수로 범위 벗어나는지 확인한다.
- 벗어나는 경우 다시 좌표값에 +1을 해서 되돌리고 맵을 벗어났다고 출력.
- 안벗어나는 경우 왼쪽으로 한칸 이동한다고 출력 후, displayMap()함수로 현재 지도 및 사용자 위치 출력 후, HP_change()함수로 체력변화, checkState()함수로 아이템, 적, 포션을 만나면 문구를 출력한다.

1. 코드블록 스크린샷

```
else if (user_input == "right") {  
    // TODO: 오른쪽으로 이동하기  
    user_x += 1;  
    bool inMap = checkXY(user_x, mapX, user_y, mapY);  
    if (inMap == false) {  
        cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;  
        user_x -= 1;  
    }  
    else {  
        cout << "오른쪽으로 이동합니다." << endl;  
        displayMap(map, user_x, user_y);  
        // 오른쪽로 이동시 HP 변화 함수  
        HP=HP_change(HP, map, user_x, user_y);  
        // 아이템, 적, 포션 만났을 때 메시지 출력 함수  
        checkState (map, user_x , user_y );  
    }  
}
```

2. 입력(블록/함수에 입력되는 변수, 값들 설명)

- user_input = 사용자 입력값
- HP = 현재 체력
- map[][] = 전체 지도
- user_x = 유저 x값
- user_y = 유저 y값
- mapY = 배열의 행 크기
- mapX = 배열의 열 크기

3. 결과(블록/함수가 종료된 결과)

- 맵을 벗어난 경우 : 기존 위치로 다시 돌아감
- 맵을 벗어나지 않는 경우 :
오른쪽으로 한칸 이동한다는 문구 출력하고, 현재 지도 출력, HP변환, 아이템, 적,
포션 만난 경우 문구 출력

4. 설명(코드 내 작동 순서, 내용 등 추가 설명)

- user_input값이 right인 경우에 조건문에 들어온다.
- 일단 user_x값을 +1하고 checkXY함수로 범위 벗어나는지 확인한다.
- 벗어나는 경우 다시 좌표값에 -1을 해서 되돌리고 맵을 벗어났다고 출력.
- 안벗어나는 경우 오른쪽으로 한칸 이동한다고 출력 후, displayMap()함수로 현재 지도 및 사용자 위치 출력 후, HP_change()함수로 체력변화, checkState()함수로 아이템, 적, 포션을 만나면 문구를 출력한다.

1. 코드블록 스크린샷

```
else if (user_input == "map") {  
    // TODO: 지도 보여주기 함수 호출  
    displayMap(map, user_x, user_y);  
}
```

2. 입력(블록/함수에 입력되는 변수, 값들 설명)

- user_input = 사용자 입력값
- map[][] = 전체 지도
- user_x = 유저 x값
- user_y = 유저 y값

3. 결과(블록/함수가 종료된 결과)

- 지도를 출력한다.

4. 설명(코드 내 작동 순서, 내용 등 추가 설명)

- 입력값이 map인 경우에 displayMap()함수로 지도를 출력해준다.

1. 코드블록 스크린샷

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

2. 입력(블록/함수에 입력되는 변수, 값들 설명)

- user_input = 사용자 입력값

3. 결과(블록/함수가 종료된 결과)

- "종료합니다" 출력 후 while문 종료.

4. 설명(코드 내 작동 순서, 내용 등 추가 설명)

- 입력값이 quit인 경우 종료합니다 출력후 break로 while문 빠져나감

1. 코드블록 스크린샷

```
else {  
    cout << "잘못된 입력입니다." << endl;  
    continue;  
}
```

2. 입력(블록/함수에 입력되는 변수, 값들 설명)

- 없음

3. 결과(블록/함수가 종료된 결과)

- 잘못된 입력이라고 출력

4. 설명(코드 내 작동 순서, 내용 등 추가 설명)

- up, down, left, right, map, quit을 제외한 다른 입력 할 시 잘못된 입력이라고 출력 후 continue로 다시 while문 반복

1. 코드블록 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(블록/함수에 입력되는 변수, 값들 설명)

- map = 전체지도
- user_x = 유저 x값
- user_y = 유저 y값

3. 결과(블록/함수가 종료된 결과)

- checkGoal 함수 리턴 값이 true인경우 목적지 도착 출력 후 종료

4. 설명(코드 내 작동 순서, 내용 등 추가 설명)

- bool타입 변수에 checkGoal함수로 목적지 도착여부 확인
- 조건문으로 도착한 경우 목적지 도착했다고 출력 후 게임 종료하며 while문을 빠져나간다.

1. 코드블록 스크린샷

```
// HP가 0 이하가 되었을때 실패 출력 조건문
if (HP==0){
    cout << "HP가 0 이하가 되었습니다. 실패했습니다."<< endl;
    break;
}
```

2. 입력(블록/함수에 입력되는 변수, 값들 설명)

- HP = 현재 체력

3. 결과(블록/함수가 종료된 결과)

- HP가 0인경우 실패 출력 후 while문 빠져나간다.

4. 설명(코드 내 작동 순서, 내용 등 추가 설명)

- 조건문으로 HP==0인경우 HP가 0이 되었다고 출력하고 break로 while문을 빠

저나간다.

4. 테스트

- 테스트 : 입력에 따른 원하는 결과나 나오는지 확인하는 과정

1. 기능 별 테스트 결과 : 요구사항 별 스크린샷

1) 지도 밖으로 나가게 되면 에러 메시지 출력

```
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): up
맵을 벗어났습니다. 다시 돌아갑니다.
```

2) up을 입력하면 좌표 이동 후 체력이 1 낮아진다.

```
현재 HP: 12 명령어를 입력하세요 (up, down, left, right, map, quit): up
위로 한 칸 올라갑니다.
|아이템|  적  |      |목적지|
-----
아이템|      |      |  적  | USER |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
현재 HP: 11 명령어를 입력하세요 (up, down, left, right, map, quit): down
```

3) down을 입력하면 좌표 이동 후 체력이 1 낮아진다.

적을 만나면 HP가 2 더 낮아지며 문구 출력

```
현재 HP: 17 명령어를 입력하세요 (up, down, left, right, map, quit): down
아래로 한 칸 내려갑니다.
|아이템|  적  |      |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      | USER | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
적이 있습니다. HP가 2 줄어듭니다.
현재 HP: 14 명령어를 입력하세요 (up, down, left, right, map, quit):
```

- 4) left를 입력하면 좌표 이동 후 체력이 1 낮아지며
포션을 먹으면 체력이 2 증가하고 문구 출력

```

현재 HP: 14 명령어를 입력하세요 (up, down, left, right, map, quit): left
왼쪽으로 이동합니다.
      |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) right를 입력하면 좌표 이동 후 체력이 1 낮아지며
아이템이 있으면 문구 출력 확인

```

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

```

- 6) map입력시 지도 출력

```

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

```

7) quit입력 시 종료합니다 출력 후 종료

```
현재 HP: 15 명령어를 입력하세요 (up, down, left, right, map, quit): quit
종료합니다.
C:\TaeunPark\CPP2409>
```

8) 이외의 입력 할 시 잘못된 입력이라고 출력

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

9) 체력이 0이 될 경우 HP가 0이하가 됐다고 출력 후 종료.

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

10) 목적지에 도착 시 도착했다는 문구 출력 후 종료.

```
현재 HP: 15 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.
```

2. 최종 테스트 스크린샷 : 프로그램 전체 동작 스크린샷

현재 HP: 20 명령어를 입력하세요 (up, down, left, right, map, quit): right
오른쪽으로 이동합니다.

	USER	적		목적지
아이템			적	
	적	포션		
포션				적

아이템이 있습니다.

현재 HP: 19 명령어를 입력하세요 (up, down, left, right, map, quit): 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, quit): down
아래로 한 칸 내려갑니다.

	아이템	적		목적지
아이템		USER	적	
	적	포션		
포션				적

현재 HP: 15 명령어를 입력하세요 (up, down, left, right, map, quit): right
오른쪽으로 이동합니다.

	아이템	적		목적지
아이템			USER	
		적	포션	
포션				적

적이 있습니다. HP가 2 줄어듭니다.

현재 HP: 12 명령어를 입력하세요 (up, down, left, right, map, quit): right
현재 HP: 12 명령어를 입력하세요 (up, down, left, right, map, quit): right
오른쪽으로 이동합니다.

	아이템	적		목적지
아이템			적	USER
		적	포션	
포션				적

현재 HP: 11 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.

C:\TaeunPark\CPP2409>

5. 결과 및 결론

1. 프로젝트 결과 : mud game 을 만들었다

2. 느낀 점 :

- 처음 과제가 주어졌을 때는 엄청 어려울 줄만 알았는데 강의에서 배운 것 바탕으로 여러 함수들을 정의하고 활용하며 코드를 완성하니까 나름 쉽게 짤 수 있었고 알찼던 것 같다. 반복되는 코드가 많은 경우 함수를 사용하면 얼마나 간편하게 코드를 짤 수 있는지 느낄 수 있었다.

- 기존 베이스코드가 주어지고 이를 바탕으로 수정하는 방식의 과제라 과제를 처음 접근하는데 큰 부담감이 없어서 좋은 것 같다. 베이스코드를 바탕으로 여러 시도를 해보며 코드를 작성할 수 있는 것 같다.

- Visual studio code에서 아직 한국어를 cin으로 입력 받으면 오류가 나오는데 이유를 잘 모르겠다. 그래서 up, down, left, right, map, quit같이 영어로 수정해서 코드를 짰다.