

**C++프로그래밍및실습**

# **9주차 프로젝트 보고서**

**MUD Game**

제출일자:

제출자명: 김태균

제출자학번: 213961

## 1. 서론

- A. 프로젝트 목적 및 배경: 9주차까지 배운 내용에 대한 실습을 위해 진행
- B. 목표: 간단한 MUD Game 구현

## 2. 요구사항

- A. 사용자 요구사항: 유저가 상하좌우로만 이동하며 목적지에 도착하는 게임
- B. 기능 계획
  - i. 게임 시작 시 초기값 세팅
  - ii. 현재 체력 출력 및 명령 입력 받기
  - iii. 상, 하, 좌, 우 명령을 입력 받았을 때 유저 위치 이동
  - iv. 유저 이동 후 이벤트 있을 경우 이벤트에 따른 결과 출력
  - v. 지도 명령 입력 시 또는 유저 이동 후 현재 지도 출력
  - vi. 종료 명령 입력 받았을 때 게임 종료
  - vii. 잘못된 명령을 받았을 경우 오류 메시지 출력
  - viii. 목적지 도착 또는 체력이 0이하가 됐을 경우 게임 종료
- C. 함수 계획
  - i. get 메서드
    - User 객체의 멤버 변수를 얻는 함수
  - ii. `bool move(int mapX, int mapY, string user_input)`
    - User 객체를 움직이게 하는 함수
  - iii. `bool checkXY(int mapX, int mapY, string user_input)`

- 이동한 후 좌표 값이 맵을 벗어나는지 체크하는 함수

iv. void displayMap(int map[][], User user)

전체 지도와 현재 유저 위치를 출력하는 함수

v. void checkState(int map[][], User user)

- 유저가 이벤트가 있는 좌표로 이동했을 때 이벤트 처리 함수

vi. void setHp(int hpDiff), void equip\_weapon(), void equip\_armor(), void lose\_armor()

- User 객체의 HP, 무기 소지 여부, 갑옷 개수 조절 함수

vii. bool checkGoal(int map[][], User user)

- 유저가 목적지에 도달했는지 여부를 반환하는 함수

### 3. 설계 및 구현

#### A. 게임 시작 시 초기값 세팅

##### i. 코드 스크린샷

```
5  const int mapX = 5;
6  const int mapY = 5;
7
8  class User{
9  private:
10     int x = 0; // 가로 번호
11     int y = 0; // 세로 번호
12     int hp = 20; // 유저 체력
13     bool weapon = false; // 무기 가지고 있는지 여부
14     int armor = 0; // 갑옷 가지고 있는지 여부
15
37 public:
38     // private 멤버 변수를 리턴하는 get 메서드
39     int getX(){
40         return x;
41     }
42     int getY(){
43         return y;
44     }
45     int getHp(){
46         return hp;
47     }
48     bool getWeapon(){
49         return weapon;
50     }
51     int getArmor(){
52         return armor;
53     }
```

```

125 // 0은 빈 공간, 1은 아이템, 2는 적, 3은 포션, 4는 목적지
126 int map[mapY][mapX] = {
127     {0, 1, 2, 0, 4},
128     {1, 0, 0, 2, 0},
129     {0, 0, 0, 0, 0},
130     {0, 2, 3, 0, 0},
131     {3, 0, 0, 0, 2}
132 };
133
134 // 유저 객체 생성
135 User user;

```

## ii. 입력

- const int mapX = 맵의 가로 크기
- const int mapY = 맵의 세로 크기
- int map[mapY][mapX] = 전체 지도
- class User = User 클래스

# int x = 유저의 x좌표 값

# int y = 유저의 y좌표 값

# int hp = 유저의 HP 값

# bool weapon = 무기 소유 여부

# int armor = 갑옷 개수

## iii. 결과

- User 클래스 정의
- User 클래스를 생성하는 것으로 게임 시작 시 초기값을 설정

## iv. 설명

- 게임 시작 시 (0, 0)에서 HP가 20인 상태로 시작
- 맵의 구성을 설정
- private 멤버 변수를 반환하는 get메서드 구현

## B. 현재 체력 출력 및 명령 입력 받기

### i. 코드 스크린샷

```
140 // 사용자의 입력을 저장할 변수
141 string user_input = "";
142
143 cout << "현재 HP: " << user.getHp() << " 명령어를 입력하세요 (상,하,좌,우,지도,종료): ";
144 cin >> user_input;
```

### ii. 입력

- string user\_input = 입력받은 유저 명령 저장

### iii. 결과

- 유저의 현재 HP와 입력 받을 수 있는 명령어 목록 출력
- 명령어를 입력 받음

### iv. 설명

- 무한 반복 while문 안에서 작동
- 입력 받은 명령어를 user\_input에 저장한다

## C. 상, 하, 좌, 우 입력을 받았을 때 유저 위치 이동

### i. 코드 스크린샷

```
146 if (user_input == "상" || user_input == "하" || user_input == "좌" || user_input == "우"){
147     // 입력이 유효해 유저가 이동했을 경우 displayMap, checkState 메서드 호출
148     if(user.move(mapX, mapY, user_input) == true){
149         displayMap(map, user);
150         checkState(map, user);
151     }
152 }
```

### ii. 입력

- string user\_input: 저장된 명령어
- int mapX, mapY: 맵 크기
- int map[mapY][mapX]: 맵 구성

- User user: 유저 객체

iii. 결과

- user객체의 move함수 호출

- displayMap 함수 호출

- checkState 함수 호출

iv. 설명

- 유저가 이동하지 못 하는 입력을 했을 경우 if문 안의 내용은 실행되지 않는다

D. move 함수

i. 함수 스크린샷

```
57 // 유저의 위치를 이동시키는 메서드
58 // 이동이 유효할 경우 true 그렇지 않을 경우 false 반환
59 bool move(int mapX, int mapY, string user_input){
60     if (user_input == "상") {
61         // 위로 한 칸 올라가기
62         this->y--;
63         bool inMap = this->checkXY(mapX, mapY, user_input);
64         if (inMap == true) {
65             this->setHp(-1);
66             cout << "위로 한 칸 올라갑니다." << endl;
67             return true;
68         }
69     }
70     else if (user_input == "하") {
71         // 아래로 한 칸 내려가기
72         this->y++;
73         bool inMap = this->checkXY(mapX, mapY, user_input);
74         if (inMap == true) {
75             this->setHp(-1);
76             cout << "아래로 한 칸 내려갑니다." << endl;
77             return true;
78         }
79     }
}
```

```

80     else if (user_input == "좌") {
81         // 왼쪽으로 한 칸 이동하기
82         this->x--;
83         bool inMap = this->checkXY(mapX, mapY, user_input);
84         if (inMap == true) {
85             this->setHp(-1);
86             cout << "왼쪽으로 이동합니다." << endl;
87             return true;
88         }
89     }
90     else if (user_input == "우") {
91         // 오른쪽으로 한 칸 이동하기
92         this->x++;
93         bool inMap = this->checkXY(mapX, mapY, user_input);
94         if (inMap == true) {
95             this->setHp(-1);
96             cout << "오른쪽으로 이동합니다." << endl;
97             return true;
98         }
99     }
100     return false;
101 }

```

## ii. 입력

- int mapX, mapY: 맵 크기
- string user\_input: 입력된 명령

## iii. 반환값

- 맵을 벗어나는 유효하지 않는 이동일 경우 false, 유효한 이동일 경우 true 반환

## iv. 결과

- checkXY 함수를 호출
- 유효한 이동일 경우 유저의 위치를 이동시키고 HP 1 감소시킨다
- 이동한 방향에 맞게 메시지 출력

## v. 설명

- User 클래스 public 접근제어자로 정의되어 있다
- 먼저 유저의 위치를 이동시킨 후 checkXY 함수를 호출해 이동이 유효한지 검사한다

- HP 감소, 메시지 출력은 값이 유효할 때만 작동한다

#### E. checkXY 함수

##### i. 함수 스크린샷

```
16 // 이동하려는 곳이 유효한 좌표인지 체크하는 함수
17 // User 객체를 참조자 매개변수로 받는다
18 bool checkXY(int mapX, int mapY, string user_input) {
19     if (this->x >= 0 && this->x < mapX && this->y >= 0 && this->y < mapY) {
20         return true;
21     } else{
22         // 맵을 벗어났을 경우 오류 메시지를 출력하고 원래 좌표로 돌아간다
23         cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
24         if(user_input == "상"){
25             this->y++;
26         } else if(user_input == "하"){
27             this->y--;
28         } else if(user_input == "좌"){
29             this->x++;
30         } else if(user_input == "우"){
31             this->x--;
32         }
33     }
34     return false;
35 }
```

##### ii. 입력

- int mapX, mapY: 맵 크기

- string user\_input: 유저 입력

##### iii. 반환값

- 맵을 벗어나는 유효하지 않는 이동일 경우 false, 유효한 이동일 경우 true 반환

##### iv. 결과

- 유효하지 않는 이동일 경우 유저 입력의 반대 방향으로 다시 이동 및 오류 메시지 출력

##### v. 설명

- User 클래스 private 접근제어자로 정의되어 있다



- 유저의 위치 좌표가 맵을 벗어나는지 조건문을 통해 판단한다

## F. displayMap 함수

### i. 함수 스크린샷

```
183 // 지도와 사용자 위치 출력하는 함수
184 void displayMap(int map[][mapX], User user) {
185     for (int i = 0; i < mapY; i++) {
186         for (int j = 0; j < mapX; j++) {
187             if (i == user.getY() && j == user.getX()) {
188                 cout << " USER |"; // 양 옆 1칸 공백
189             }
190             else {
191                 int posState = map[i][j];
192                 switch (posState) {
193                     case 0:
194                         cout << "      |"; // 6칸 공백
195                         break;
196                     case 1:
197                         cout << "아이템|";
198                         break;
199                     case 2:
200                         cout << " 적   |"; // 양 옆 2칸 공백
201                         break;
202                     case 3:
203                         cout << " 포션 |"; // 양 옆 1칸 공백
204                         break;
205                     case 4:
206                         cout << "목적지|";
207                         break;
208                 }
209             }
210         }
211         cout << endl;
212         cout << " ----- " << endl;
213     }
214 }
```

### ii. 입력

- int map[][] = 전체 지도
- User user = 유저 객체

### iii. 반환값

- 없음

### iv. 결과

- 전체 지도 출력

- 현재 유저 위치 출력

#### v. 설명

- 전역 함수로 정의되어 있다
- 2차원 배열에 있는 맵을 출력하다가 유저 위치와 동일한 좌표를 발견할 경우 유저 출력

### G. checkState 함수

#### i. 함수 스크린샷

```

226 // 아이템, 포션, 적을 만났을 때 메시지 출력
227 // User 객체를 참조자 매개변수로 받는다
228 void checkState(int map[][mapX], User& user){
229     if(map[user.getY()][user.getX()] == 1){
230         // 아이템 중 무기와 갑옷을 랜덤하게 얻는다
231         int item = rand() % 2;
232         if(item == 0){
233             user.equip_weapon();
234             cout << "무기를 얻었습니다." << endl;
235         } else{
236             user.equip_armor();
237             cout << "갑옷을 얻었습니다. 현재 갑옷 개수: " << user.getArmor() << endl;
238         }
239     } else if(map[user.getY()][user.getX()] == 2){
240         cout << "적이 있습니다. ";
241         if(user.getArmor() > 0){
242             // 갑옷이 있을 경우 HP가 줄어들지 않고 갑옷의 개수가 하나 줄어든다
243             // 갑옷과 무기가 둘 다 있을 경우 갑옷을 먼저 소모하도록 설계
244             user.lose_armor();
245             cout << "갑옷으로 막아내 피해를 막았습니다. 갑옷이 파괴됩니다. 현재 갑옷 개수: " << user.getArmor() << endl;
246         } else if(user.getWeapon() == true){
247             // 무기가 있을 경우 HP가 1만 줄어든다
248             cout << "무기가 있습니다. HP가 1 줄어듭니다." << endl;
249             user.setHp(-1);
250         } else{
251             // 갑옷과 무기 둘 다 없을 경우 HP가 2 줄어든다
252             cout << "HP가 2 줄어듭니다." << endl;
253             user.setHp(-2);
254         }
255     } else if(map[user.getY()][user.getX()] == 3){
256         cout << "포션이 있습니다. HP가 2 회복됩니다." << endl;
257         user.setHp(2);
258     }
259 }

```

#### ii. 입력

- int map[][] = 전체 지도
- User& user = 유저 객체 참조자 변수

iii. 반환값

- 없음

iv. 결과

- 현재 유저가 위치한 곳이 아이템, 포션, 적 이벤트일 경우 해당 이벤트 처리

# 아이템: equip\_weapon, equip\_armor 중 랜덤하게 호출

# 포션: setHp 호출, HP 2 증가

# 적: 무기, 갑옷 소지 여부에 따라 setHp, lose\_armor 호출

- 이벤트 처리에 따른 메시지 출력

v. 설명

- 전역 함수로 정의되어 있다
- 아이템을 얻을 경우 무기와 갑옷 중 랜덤하게 얻도록 설계
- 포션을 얻을 경우 HP가 회복되도록 설계
- 적을 만났을 경우

# 갑옷이 있을 경우: 피해를 입지 않고 갑옷이 하나 파괴

# 갑옷이 없고 무기가 있을 경우: HP 1 감소

# 갑옷이 없고 무기가 없을 경우: HP 2 감소

H. setHp, equip\_weapon, equip\_armor, lose\_armor 함수

i. 함수 스크린샷

```

101     void setHp(int hpDiff){
102         // HP 변경 전/후 차이를 매개변수로 받는다
103         this->hp += hpDiff;
104     }
105     void equip_weapon(){
106         this->weapon = true;
107     }
108     void equip_armor(){
109         this->armor++;
110     }
111     void lose_armor(){
112         this->armor--;
113     }

```

## ii. 입력

- setHp

# int hpDiff = HP 변경 전후 차이

- equip\_weapon, equip\_armor, lose\_armor

# 없음

## iii. 반환값

- 없음

## iv. 결과

- setHp: hpDiff만큼 hp변경

- equip\_weapon: weapon 값 true로 변경

- equip\_armor: armor 값 1 증가

- lose\_armor: armor 값 1 감소

## v. 설명

- User 클래스 public 접근제어자로 정의되어 있다

## I. 지도 명령 입력 시 전체 지도 및 유저 위치 출력

### i. 코드 스크린샷

```

152     else if (user_input == "지도") {
153         // TODO: 지도 보여주기 함수 호출
154         displayMap(map, user);
155     }

```

ii. 입력

- string user\_input = 입력된 명령

iii. 결과

- displayMap 함수 호출

iv. 설명

- D. 기능과 else if로 연결

- 입력된 명령이 지도일 경우 displayMap 함수를 호출해 전체 지도와 유저의 위치를 출력한다

J. 종료 명령 입력 받았을 때 게임 종료

i. 코드 스크린샷

```

156     else if (user_input == "종료") {
157         cout << "종료합니다.";
158         break;
159     }

```

ii. 입력

- string user\_input = 입력된 명령

iii. 결과

- 종료 메시지 출력

- break문을 통해 while문 탈출

iv. 설명

- I. 기능과 else if로 연결

- while문을 탈출하는 것으로 게임 종료

K. 잘못된 명령 입력 받았을 경우 오류 메시지 출력

i. 코드 스크린샷

```
160     else {
161         cout << "잘못된 입력입니다." << endl;
162         continue;
163     }
```

ii. 입력

- 없음

iii. 결과

- 오류 메시지 출력

- continue 실행

iv. 설명

- J. 기능과 else로 연결

L. 목적지 도착 또는 체력이 0이하가 됐을 경우 게임 종료

i. 코드 스크린샷

```
165     // 유저의 hp가 0 이하인지 체크
166     if(user.getHp() <= 0){
167         cout << "hp가 0이하가 되었습니다. 실패했습니다." << endl;
168         cout << "게임을 종료합니다." << endl;
169         break;
170     }
171
172     // 목적지에 도달했는지 체크
173     bool finish = checkGoal(map, user);
174     if (finish == true) {
175         cout << "목적지에 도착했습니다! 축하합니다!" << endl;
176         cout << "게임을 종료합니다." << endl;
177         break;
178     }
```

ii. 입력

- user.getHP() = 유저의 현재 HP

- int map[][] = 전체 지도

- User user = 유저 객체

- bool finish = 목적지 도달 여부

### iii. 결과

- 체력이 0이하일 경우 while문 탈출

- checkGoal 함수를 호출해 결과 값을 finish에 저장

- finish값에 따라 while문 탈출

- while문을 탈출할 경우 메시지 출력

### iv. 설명

- finish 값이 true일 경우 while문을 탈출한다

- while문을 탈출하는 것으로 게임종료

## M. checkGoal 함수

### i. 함수 스크린샷

```
217 // 유저의 위치가 목적지인지 체크하는 함수
218 // User 객체를 매개변수로 받아 x, y좌표를 구한다
219 bool checkGoal(int map[][mapX], User user) {
220     // 목적지 도착하면
221     if (map[user.getY()][user.getX()] == 4) {
222         return true;
223     }
224     return false;
225 }
```

### ii. 입력

- int map[][] = 전체 지도

- User user = 유저 객체

iii. 반환값

- 유저가 목적지에 도착했을 경우 true, 아닐 경우 false 반환

iv. 결과

- 유저의 목적지 도착여부를 반환한다

v. 설명

- 전역함수로 정의되어 있다



## 4. 테스트

### A. 게임 시작 시 초기값 설정

```
(base) taegyun-ui-MacBookAir:~ rbsxo135$ /Users/rbsxo135/Desktop/C++2409/project/week9/mud_game
; exit;
현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): █
```

### B. 상, 하, 좌, 우 이동 명령 시 이동, 전체 지도 및 현재 위치 출력, 체력 감소

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

아이템		적	목적지	
아이템	USER		적	
	적	포션		
포션				적

현재 HP: 18 명령어를 입력하세요 (상,하,좌,우,지도,종료): 하  
아래로 한 칸 내려갑니다.

아이템		적	목적지	
아이템			적	
	USER			
	적	포션		
포션				적

현재 HP: 17 명령어를 입력하세요 (상,하,좌,우,지도,종료): 상  
위로 한 칸 올라갑니다.

아이템		적	목적지	
아이템	USER		적	
	적	포션		
포션				적

현재 HP: 15 명령어를 입력하세요 (상,하,좌,우,지도,종료): 좌  
왼쪽으로 이동합니다.

아이템		적	목적지	
아이템	USER		적	
	적	포션		
포션				적

### C. 맵을 벗어났을 경우 원래 위치로 복귀

USER	아이템	적	목적지
아이템		적	
	적	포션	
포션			적

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): █

### D. 유저 이동 후 아이템 이벤트 실행

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

USER	적	목적지
아이템		적
	적	포션
포션		적

갑옷을 얻었습니다. 현재 갑옷 개수: 1

현재 HP: 14 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우  
 오른쪽으로 이동합니다.

USER	적	목적지
아이템		적
	적	포션
포션		적

무기를 얻었습니다.

### E. 유저 이동 후 포션 이벤트 실행

현재 HP: 10 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우  
 오른쪽으로 이동합니다.

아이템	적	목적지
아이템		적
	적	USER
포션		적

포션이 있습니다. HP가 2 회복됩니다.

현재 HP: 11 명령어를 입력하세요 (상,하,좌,우,지도,종료): █

## F. 유저 이동 후 적 이벤트 실행

### i. 무기가 있는 경우

왼 쪽 으 로 이 동 합 니 다 .

아 이 템   적		목 적 지	
아 이 템		적	
-----			
-----			
	USER	포 셴	
-----			
포 셴			적

적 이 있 습 니 다 . 갑 옷 으 로 막 아 내 피 해 를 막 았 습 니 다 . 갑 옷 이 파 괴 됩 니 다 . 현 재 갑 옷 개 수 : 1  
현 재 HP: 10 명 령 어 를 입 력 하 세 요 (상 ,하 ,좌 ,우 ,지 도 ,종 료 ): 우  
오 른 쪽 으 로 이 동 합 니 다 .

아 이 템   적		목 적 지	
아 이 템		적	
-----			
-----			
	적	USER	
-----			
포 셴			적

포 셴 이 있 습 니 다 . HP가 2 회 복 됩 니 다 .  
현 재 HP: 11 명 령 어 를 입 력 하 세 요 (상 ,하 ,좌 ,우 ,지 도 ,종 료 ): 좌  
왼 쪽 으 로 이 동 합 니 다 .

아 이 템   적		목 적 지	
아 이 템		적	
-----			
-----			
	USER	포 셴	
-----			
포 셴			적

적 이 있 습 니 다 . 갑 옷 으 로 막 아 내 피 해 를 막 았 습 니 다 . 갑 옷 이 파 괴 됩 니 다 . 현 재 갑 옷 개 수 : 0  
현 재 HP: 10 명 령 어 를 입 력 하 세 요 (상 ,하 ,좌 ,우 ,지 도 ,종 료 ): 우  
오 른 쪽 으 로 이 동 합 니 다 .

아 이 템   적		목 적 지	
아 이 템		적	
-----			
-----			
	적	USER	
-----			
포 셴			적

포 셴 이 있 습 니 다 . HP가 2 회 복 됩 니 다 .  
현 재 HP: 11 명 령 어 를 입 력 하 세 요 (상 ,하 ,좌 ,우 ,지 도 ,종 료 ): 좌  
왼 쪽 으 로 이 동 합 니 다 .

아 이 템   적		목 적 지	
아 이 템		적	
-----			
-----			
	USER	포 셴	
-----			
포 셴			적

적 이 있 습 니 다 . 무 기 가 있 습 니 다 . HP가 1 줄 어 듭 니 다 .  
현 재 HP: 9 명 령 어 를 입 력 하 세 요 (상 ,하 ,좌 ,우 ,지 도 ,종 료 ): ■

## ii. 무기가 없는 경우

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

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

적이 있습니다. 갑옷으로 막아내 피해를 막았습니다. 갑옷이 파괴됩니다. 현재 갑옷 개수: 0

현재 HP: 18 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우  
오른쪽으로 이동합니다.

	아 이 템	적		USER		목 적 지	
-----							
아 이 템				적			
-----							
-----							
		적		포 셴			
-----							
포 셴						적	
-----							

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

아이템		USER		목적지	
아이템			적		
	적	포션			
포션					적

적이 있습니다. HP가 2 줄어듭니다.

현재 HP: 14 명령어를 입력하세요 (상,하,좌,우,지도,종료):

## G. 지도 명령 시 전체 지도 및 현재 위치 출력

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

USER	아이템	적	목적지
아이템		적	
	적	포션	
포션			적

## H. 종료 명령 시 게임 종료

현재 HP: 14 명령어를 입력하세요 (상,하,좌,우,지도,종료): 종료  
종료합니다.logout

## I. 잘못된 명령 입력할 경우 오류 메시지 출력

현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): 점프  
잘못된 입력입니다.

## J. 목적지에 도착했을 경우 게임 종료

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

아이템		적	USER	
아이템		적		
	적	포션		
포션			적	

목적지에 도착했습니다! 축하합니다!  
게임을 종료합니다.  
logout

## K. 체력이 0이 되었을 경우 게임 종료

현재 HP: 1 명령어를 입력하세요 (상,하,좌,우,지도,종료): 좌  
왼쪽으로 이동합니다.

아이템		적	목적지	
아이템	USER	적		
	적	포션		
포션			적	

hp가 0이하가 되었습니다. 실패했습니다.  
게임을 종료합니다.  
logout

## 5. 결과 및 결론

A. 프로젝트 결과: MUD Game을 만들었다

B. 느낀 점: User 클래스를 정의하여 게임을 만드는 과정에서 main 함수가  
간결해지고 객체의 상호작용을 통해 프로그램이 진행되는 것을 알 수  
있었다