Mud game

과목: C++프로그래밍및실습

학과: 소프트웨어공학과

학번: 214270

이름: 현경서

목차

1. 서론

1) 프로젝트 및 배경

9주차까지 배운 내용에 대한 실습

2) 목표

mud game 구현

2. 요구사항

1) 사용자 요구사항

사용자가 게임 내 유저를 이동시켜서 목적지에 도달하기

2) 기능 요구사항

① 사용자에게 "상", "하", "좌", "우", "지도", "종료" 중 하나를 입력 받기
- 입력 받을 때 마다 HP 함께 출력
- 상/하/좌/우 입력 시 해당 방향으로 이동
- "지도" 입력 시 전체 지도와 함께 현재 위치 출력
- "종료" 입력 시 프로그램 종료
- 이 중 다른 것을 입력하면 에러 메시지 출력 후 재 입력 요청

② 지도 출력

③ 이동
- 유저가 이동할 때 마다 HP 1씩 감소
- 이동 후 지도 출력

④ 이동 유효성 검사

⑤ 지도 밖에 나가게 되면 에러 메시지 출력

⑥ 이벤트 발생 시 그에 대한 메시지 출력 및 HP 변화
- 아이템: HP 변화 없음
- 포션: HP + 2
- 적: HP - 2

⑦ 목적지에 도착하면 "성공"을 출력하고 종료

⑧ HP가 0이 되면 "실패"를 출력하고 종료

3) 클래스 계획

① User: 유저 기본 정보

- 멤버 변수 (private)
  - user_x, user_y: 유저의 현재 위치
  - hp: 유저의 현재 HP
- 멤버 메소드 (public)
  - getUser_x(), getUser_y(), getHp(): 멤버 변수 조회
  - Move(): 유저의 이동
  - ChangeHp(): 유저의 HP 변화

4) 함수 계획
   ① main(): 사용자에게 값을 계속 입력 받고 그에 대한 함수 호출
   ② displayMap(): 지도와 현재 위치 출력 함수
   ③ checkXY(): 사용자 위치 체크 함수
   ④ checkGoal(): 목적지 도착 체크 함수
   ⑤ checkState(): 이벤트 발생 체크 함수

3. 설계 및 구현

1) 기능 별 요구사항
   ① 사용자에게 "상", "하", "좌", "우", "지도", "종료" 중 하나를 입력 받기

```
50      // 사용자의 입력을 저장할 변수
51      string user_input = "";
52
53      // 이동 거리 변수
54      int move_x = 0;
55      int move_y = 0;
56
57      // 입력
58      cout << "현재 HP: " << user1.getHp();
59      cout << "  명령어를 입력하세요 (상,하,좌,우,지도,종료): ";
60      cin >> user_input;
61
62      if (user_input == "상") move_y = -1;
63      else if (user_input == "하") move_y = 1;
64      else if (user_input == "좌") move_x = -1;
65      else if (user_input == "우") move_x = 1;
66      else if (user_input == "지도") {
67          // TODO: 지도 보여주기 함수 호출
68          displayMap(map, user1.getUser_x(), user1.getUser_y());
69          continue;
70      }
71      else if (user_input == "종료") {
72          cout << "종료합니다.";
73          break;
74      }
75      else {
76          cout << "잘못된 입력입니다." << endl;
77          continue;
78      }
```

I. 입력
- 사용자의 입력

II. 결과
- 입력이 상/하/좌/우인 경우 이동 정보를 저장하는 변수인 move_x, move_y에 이동 방향에 따른 이동 거리를 저장
- 입력이 "지도"인 경우 displayMap() 함수 호출을 통한 지도 출력
- 입력이 "종료"인 경우 프로그램 종료
- 그 외의 입력의 경우 재입력 요청

III. 설명
- else if문을 통해 사용자의 입력을 분류하고 그 결과에 따른 명령을 수행한다.

② 지도 출력

```
132  // 지도와 사용자 위치 출력하는 함수
133  void displayMap(int map[][mapX], int user_x, int user_y) {
134      for (int i = 0; i < mapY; i++) {
135          for (int j = 0; j < mapX; j++) {
136              if (i == user_y && j == user_x) {
137                  cout << " USER |"; // 양 옆 1칸 공백
138              }
139              else {
140                  int posState = map[i][j];
141                  switch (posState) {
142                  case 0:
143                      cout << "      |"; // 6칸 공백
144                      break;
145                  case 1:
146                      cout << "아이템|";
147                      break;
148                  case 2:
149                      cout << "  적  |"; // 양 옆 2칸 공백
150                      break;
151                  case 3:
152                      cout << " 포션 |"; // 양 옆 1칸 공백
153                      break;
154                  case 4:
155                      cout << "목적지|";
156                      break;
157                  }
158              }
159          }
160          cout << endl;
161          cout << " ------------------------------ " << endl;
162      }
163  }
```

I. 입력
- map[][]: 맵
- user_x: 유저의 현재 x 위치
- user_y: 유저의 현재 y 위치

II. 결과
- 지도와 유저의 현재 위치를 출력

III. 설명
- 중첩 for문과 switch - case문을 통해 지도에 현재 유저 위치, 아이템, 적, 포션을 포함하여 출력한다.

③ 이동

```
80          // 이동
81          user1.Move(move_x, move_y);
```

```
21      void Move(int move_x, int move_y){
22          user_x += move_x;
23          user_y += move_y;
24      }
```

I. 입력
- move_x: x 방향 이동 거리
- move_y: y 방향 이동 거리

II. 결과
- 유저의 현재 위치 변경

III. 설명
- 멤버 메소드 Move()를 통해 현재 유저 위치에 이동 거리를 더한다.

④ 이동 유효성 검사

```
    bool inMap = checkXY(user1.getUser_x(), mapX, user1.getUser_y(), mapY);
165 // 이동하려는 곳이 유효한 좌표인지 체크하는 함수
166 bool checkXY(int user_x, int mapX, int user_y, int mapY) {
167     bool checkFlag = false;
168     if (user_x >= 0 && user_x < mapX && user_y >= 0 && user_y < mapY) {
169         checkFlag = true;
170     }
171     return checkFlag;
172 }
```

I. 입력
- user_x: 유저의 현재 x 위치
- mapX: 맵의 x 길이
- user_y: 유저의 현재 y 위치
- mapY: 맵의 y 길이

II. 결과
- checkFlag: 이동한 위치가 유효한지 알려주는 bool형 변수

III. 설명
- 이동한 위치가 맵을 벗어나는지 if문을 통해 확인하고 결과를 반환한다.
- main() 함수에서는 반환된 값을 inMap 변수에 저장하고 이를 이용해서 유효성 확인 결과를 처리한다.
- main() 함수에서 checkXY() 함수로 인자를 보낼 때 getUser_x() 메소드와 getUser_y() 메소드를 통해 유저의 위치 정보를 전달한다.

⑤ 지도 밖에 나가게 되면 에러 메시지 출력

```
83          if (inMap == false){
84              cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
85              user1.Move(-move_x, -move_y);
86              continue;
87          }
88          else{
89              cout << user_input << "(으)로 이동합니다." << endl;
90              user1.ChangeHp(-1);
91              displayMap(map, user1.getUser_x(), user1.getUser_y());
92          }
```

I. 입력
- inMap: 이동 유효성 검사 결과

II. 결과
- inMap == false인 경우 이전 위치로 복귀
- inmap == true인 경우 HP를 감소시키고 지도 출력

III. 설명
- inMap == false인 경우에는 Moev() 메소드에 move_x, move_y 값의 음수를 인자로 전달하여 다시 이전 위치로 되돌아가게 한다.
- inMap == true인 경우에는 ChangeHp() 메소드를 통해 유저의 HP를 1 감소시키고 displayMap() 함수를 통해 지도를 출력한다.

⑥ 이벤트 발생 시 그에 대한 메시지 출력 및 HP 변화

```
        // 이벤트가 발생하는지 체크
        int event = checkState(map, user1.getUser_x(), user1.getUser_y());
        switch (event){
            case 0: // 발생 안함
                break;
            case 1: // 아이템
                cout << "아이템이 있습니다." << endl;
                break;
            case 2: // 적
                cout << "적이 있습니다. HP가 2 줄어듭니다." << endl;
                user1.ChangeHp(-2);
                break;
            case 3: // 포션
                cout << "포션이 있습니다. HP가 2 늘어납니다." << endl;
                user1.ChangeHp(2);
                break;
        }
183     // 유저의 위치가 아이템, 포션, 적인지 체크하는 함수
184     // 1은 아이템, 2는 적, 3은 포션
185     int checkState(int map[][mapX], int user_x, int user_y) {
186         if (map[user_y][user_x] == 1) return 1;
187         else if (map[user_y][user_x] == 2) return 2;
188         else if (map[user_y][user_x] == 3) return 3;
189         else return 0;
190     }
```

I. 입력

- map[][]: 맵
- user_x: 유저의 현재 x 위치
- user_y: 유저의 현재 y 위치

II. 결과

- 현재 위치에 적이 있을 경우 HP 감소
- 현재 위치에 포션이 있을 경우 HP 증가

III. 설명

- checkState() 함수는 else if문을 통하여 현재 위치에서 어떤 이벤트가 발생하는지 확인하고 이에 따른 값을 반환한다.
- 아이템이 있을 경우 1, 적이 있을 경우 2, 포션이 있을 경우 3, 아무것도 없을 경우 0을 반환한다.
- main() 함수에서 checkState() 함수를 통해 반환된 값을 event 변수에 저장하고 switch case문을 통해 event 값의 경우를 나누어서 그 다음 명령을 수행한다.
- main() 함수에서 checkState() 함수로 인자를 보낼 때 getUser_x() 메소드와 getUser_y() 메소드를 통해 유저의 위치 정보를 전달한다.
- event == 0 (아무것도 없음) 인 경우 아무 명령도 수행하지 않는다.
- event == 1 (아이템) 인 경우 아이템이 있다는 메시지를 출력한다.
- event == 2 (적) 인 경우 적이 있다는 메시지를 출력하고 ChangeHp() 메소드를 통해 유저의 HP를 2 감소시킨다.
- event == 3 (포션) 인 경우 포션이 있다는 메시지를 출력하고 ChangeHp() 메소드를 통해 유저의 HP를 2 증가시킨다.

⑦ 목적지에 도착하면 "성공"을 출력하고 종료

```
    // 목적지에 도달했는지 체크
    bool finish = checkGoal(map, user1.getUser_x(), user1.getUser_y());
    if (finish == true) {
        cout << "목적지에 도착했습니다! 축하합니다!" << endl;
        cout << "게임을 종료합니다." << endl;
        break;
    }
174 // 유저의 위치가 목적지인지 체크하는 함수
175 bool checkGoal(int map[][mapX], int user_x, int user_y) {
176     // 목적지 도착하면
177     if (map[user_y][user_x] == 4) {
178         return true;
179     }
180     return false;
181 }
```

I. 입력

- map[][]: 맵
- user_x: 유저의 현재 x 위치
- user_y: 유저의 현재 y 위치

II. 결과

- 유저의 현재 위치가 목적지일 경우 프로그램 종료

III. 설명

- checkGoal() 함수는 if문을 통해 목적지 도달 여부를 반환한다.
- main() 함수에서 checkGoal() 함수를 통해 반환된 값을 finish 변수에 저장하고 if문을 통해 finish 값에 따른 다음 명령을 수행한다.
- main() 함수에서 checkGoal() 함수로 인자를 보낼 때 getUser_x() 메소드와 getUser_y() 메소드를 통해 유저의 위치 정보를 전달한다.
- finish == true인 경우 목적지에 도달했다는 메시지를 출력하고 프로그램을 종료한다.
- finish == false인 경우 아무 명령도 수행하지 않는다.

⑧ HP가 0이 되면 "실패"를 출력하고 종료

```
120         // hp가 0이면 실패
121         if (user1.getHp() <= 0) {
122             cout << "HP가 0 이하가 되었습니다. 실패했습니다." << endl;
123             cout << "게임을 종료합니다.";
124             break;
125         }
126     }
```

I. 입력

- user1.getHp(): 유저의 HP

II. 결과

- 유저의 HP가 0보다 작거나 같으면 프로그램 종료

III. 설명

- if문을 통해 유저의 HP가 0보다 작거나 같은지 확인하고 만약 그렇다면 실패를 출력하고 프로그램을 종료한다.

4. 테스트

1) 기능 별 테스트 결과

① 사용자에게 "상", "하", "좌", "우", "지도", "종료" 중 하나를 입력 받기

- 입력 받을 때 마다 HP 함께 출력

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

- "종료" 입력 시 프로그램 종료

```
현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): 종료
종료합니다.
PS C:\code\CPP2409>
```

- 이 중 다른 것을 입력하면 에러 메시지 출력 후 재 입력 요청

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

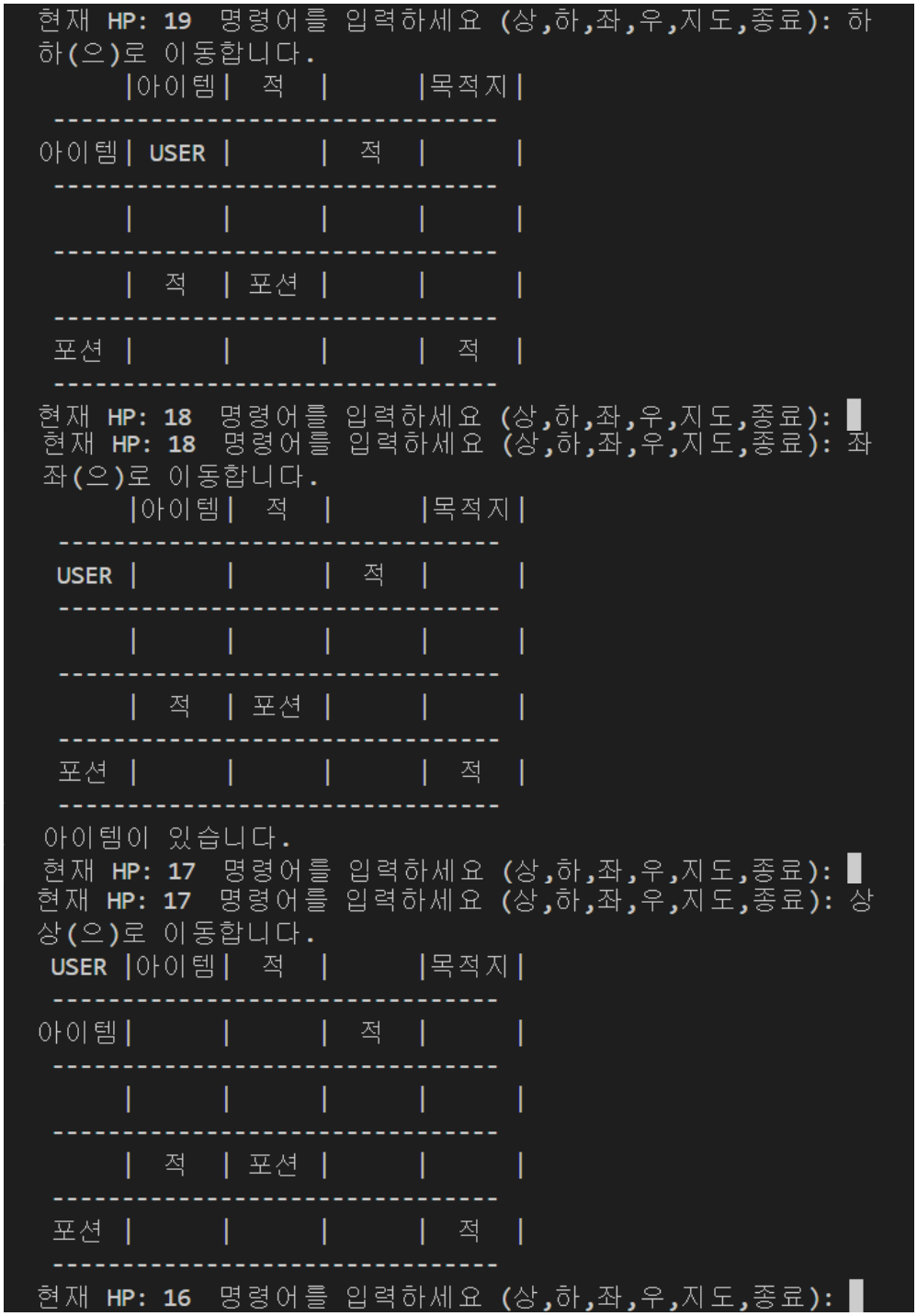
② 지도 출력

```
현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): 지도
 USER |아이템|  적  |      |목적지|
 -------------------------------
아이템|      |      |  적  |      |
 -------------------------------
      |      |      |      |      |
 -------------------------------
      |  적  | 포션 |      |      |
 -------------------------------
 포션 |      |      |      |  적  |
 -------------------------------
현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료):
```

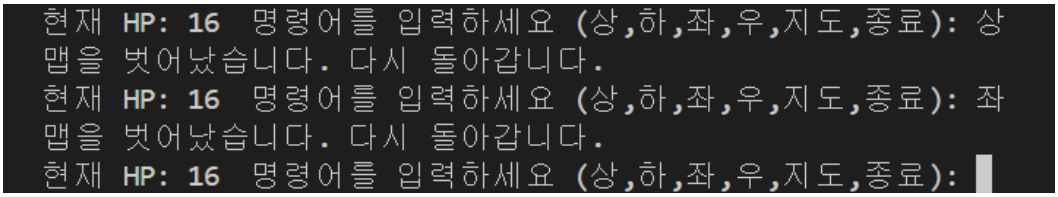
③ 이동

- 유저가 이동할 때 마다 HP 1씩 감소
- 이동 후 지도 출력

```
현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
우(으)로 이동합니다.
      | USER |  적  |      |목적지|
 -------------------------------
아이템|      |      |  적  |      |
 -------------------------------
      |      |      |      |      |
 -------------------------------
      |  적  | 포션 |      |      |
 -------------------------------
 포션 |      |      |      |  적  |
 -------------------------------
아이템이 있습니다.
현재 HP: 19 명령어를 입력하세요 (상,하,좌,우,지도,종료):
```

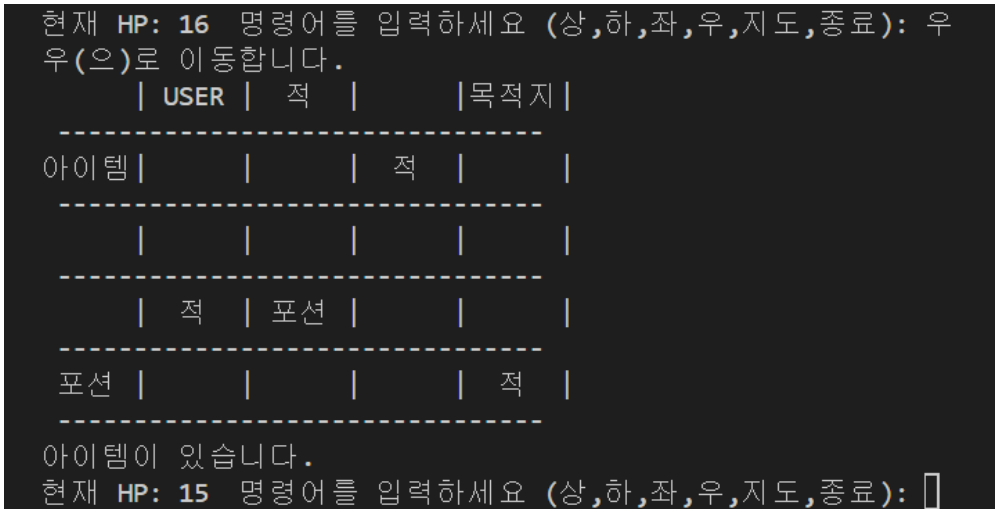


④ 지도 밖에 나가게 되면 에러 메시지 출력

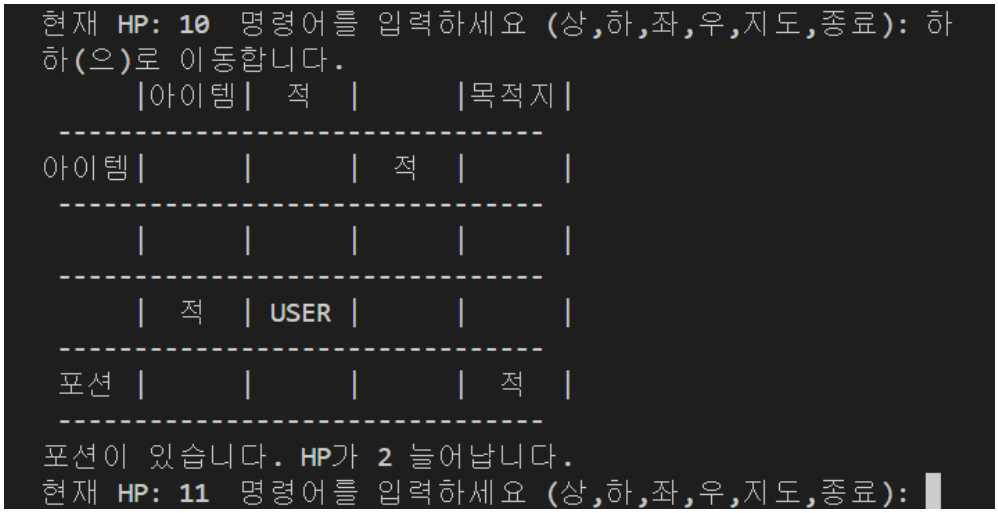


⑤ 이벤트 발생 시 그에 대한 메시지 출력 및 HP 변화

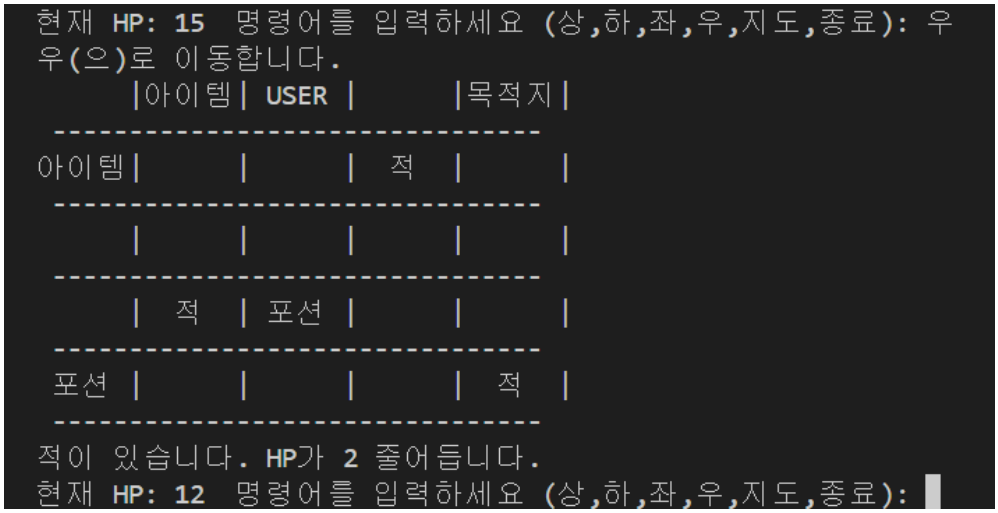
- 아이템: HP 변화 없음



- 포션: HP + 2



- 적: HP – 2



⑥ 목적지에 도착하면 "성공"을 출력하고 종료

```
현재 HP: 15  명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
우(으)로 이동합니다.
      |아이템|  적   |      | USER |
 ---------------------------------
아이템|      |      |  적   |      |
 ---------------------------------
      |      |      |      |      |
 ---------------------------------
      |  적   | 포션 |      |      |
 ---------------------------------
 포션 |      |      |      |  적   |
 ---------------------------------
목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.
PS C:\code\CPP2409>
```

⑦ HP가 0이 되면 "실패"를 출력하고 종료

```
현재 HP: 1  명령어를 입력하세요 (상,하,좌,우,지도,종료): 상
상(으)로 이동합니다.
      |아이템|  적   |      |목적지|
 ---------------------------------
아이템|      |      |  적   |      |
 ---------------------------------
      |      |      |      | USER |
 ---------------------------------
      |  적   | 포션 |      |      |
 ---------------------------------
 포션 |      |      |      |  적   |
 ---------------------------------
HP가 0 이하가 되었습니다. 실패했습니다.
게임을 종료합니다.
PS C:\code\CPP2409>
```

2) 최종 테스트 스크린샷

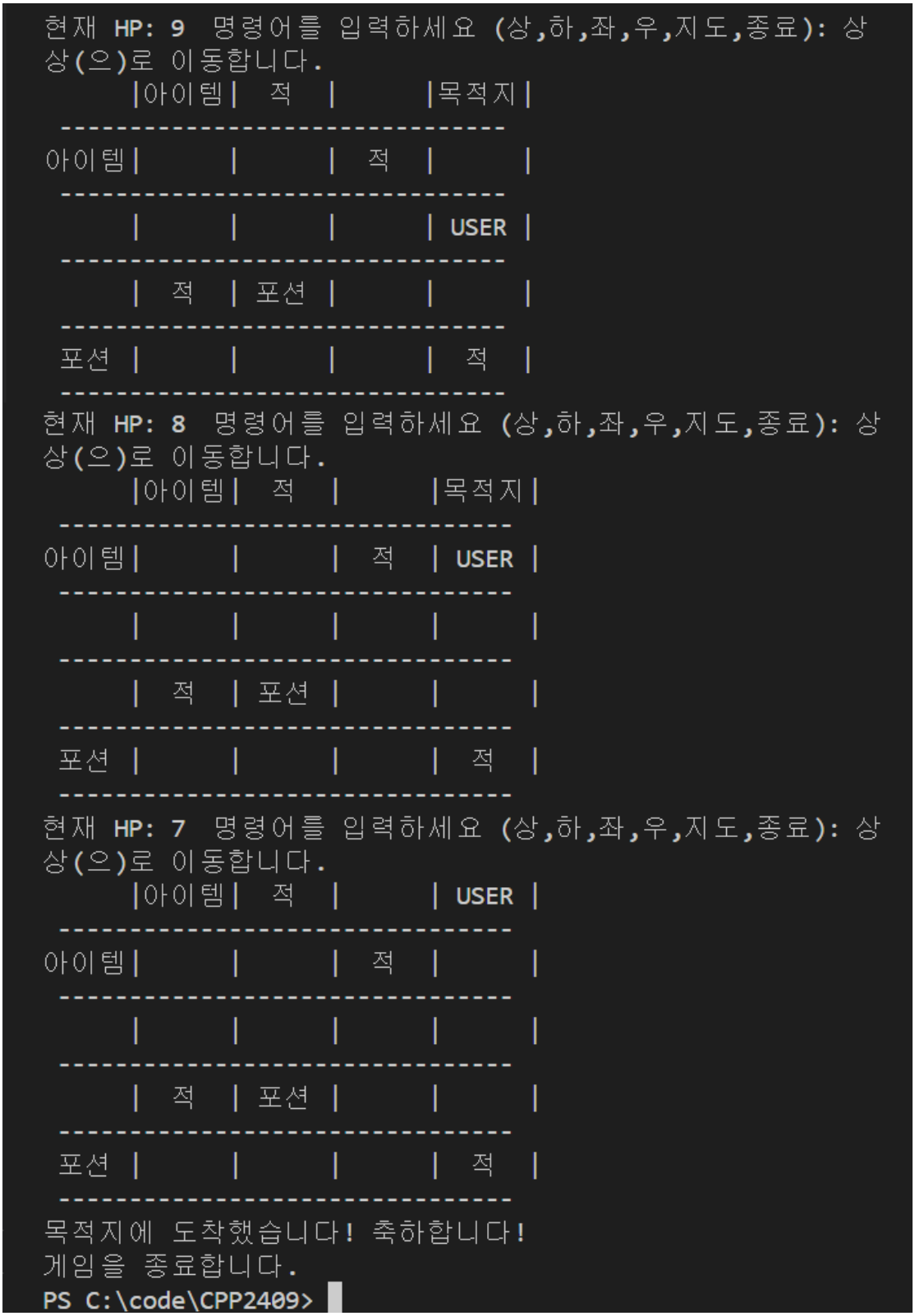
```
현재 HP: 20  명령어를 입력하세요 (상,하,좌,우,지도,종료): 지도
 USER |아이템|  적   |      |목적지|
 ---------------------------------
아이템|      |      |  적   |      |
 ---------------------------------
      |      |      |      |      |
 ---------------------------------
      |  적   | 포션 |      |      |
 ---------------------------------
 포션 |      |      |      |  적   |
 ---------------------------------
```

```
현재 HP: 20  명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
우(으)로 이동합니다.
      | USER |  적  |      |목적지|
 --------------------------------
아이템|      |      |  적  |      |
 --------------------------------
      |      |      |      |      |
 --------------------------------
      |  적  | 포션 |      |      |
 --------------------------------
 포션 |      |      |      |  적  |
 --------------------------------
아이템이 있습니다.
현재 HP: 19  명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
우(으)로 이동합니다.
      |아이템| USER |      |목적지|
 --------------------------------
아이템|      |      |  적  |      |
 --------------------------------
      |      |      |      |      |
 --------------------------------
      |  적  | 포션 |      |      |
 --------------------------------
 포션 |      |      |      |  적  |
 --------------------------------
적이 있습니다. HP가 2 줄어듭니다.
현재 HP: 16  명령어를 입력하세요 (상,하,좌,우,지도,종료): 하
하(으)로 이동합니다.
      |아이템|  적  |      |목적지|
 --------------------------------
아이템|      | USER |  적  |      |
 --------------------------------
      |      |      |      |      |
 --------------------------------
      |  적  | 포션 |      |      |
 --------------------------------
 포션 |      |      |      |  적  |
 --------------------------------
```

```
현재 HP: 15  명령어를 입력하세요 (상,하,좌,우,지도,종료): 하
하(으)로 이동합니다.
      |아이템|  적  |      |목적지|
 --------------------------------
아이템|      |      |  적  |      |
 --------------------------------
      |      | USER |      |      |
 --------------------------------
      |  적  | 포션 |      |      |
 --------------------------------
 포션 |      |      |      |  적  |
 --------------------------------
현재 HP: 14  명령어를 입력하세요 (상,하,좌,우,지도,종료): 하
하(으)로 이동합니다.
      |아이템|  적  |      |목적지|
 --------------------------------
아이템|      |      |  적  |      |
 --------------------------------
      |      |      |      |      |
 --------------------------------
      |  적  | USER |      |      |
 --------------------------------
 포션 |      |      |      |  적  |
 --------------------------------
포션이 있습니다. HP가 2 늘어납니다.
현재 HP: 15  명령어를 입력하세요 (상,하,좌,우,지도,종료): 하
하(으)로 이동합니다.
      |아이템|  적  |      |목적지|
 --------------------------------
아이템|      |      |  적  |      |
 --------------------------------
      |      |      |      |      |
 --------------------------------
      |  적  | 포션 |      |      |
 --------------------------------
 포션 |      | USER |      |  적  |
 --------------------------------
현재 HP: 14  명령어를 입력하세요 (상,하,좌,우,지도,종료): 하
맵을 벗어났습니다. 다시 돌아갑니다.
```

```
현재 HP: 14  명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
우(으)로 이동합니다.
      |아이템|  적   |      |목적지|
 --------------------------------
아이템|      |      |  적   |      |
 --------------------------------
      |      |      |       |      |
 --------------------------------
      |  적   | 포션 |       |      |
 --------------------------------
```

```
현재 HP: 13  명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
우(으)로 이동합니다.
      |아이템|  적   |      |목적지|
 --------------------------------
아이템|      |      |  적   |      |
 --------------------------------
      |      |      |       |      |
 --------------------------------
      |  적   | 포션 |       |      |
 --------------------------------
 포션 |      |      |       | USER |
 --------------------------------
적이 있습니다. HP가 2 줄어듭니다.
현재 HP: 10  명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
맵을 벗어났습니다. 다시 돌아갑니다.
현재 HP: 10  명령어를 입력하세요 (상,하,좌,우,지도,종료): 상
상(으)로 이동합니다.
      |아이템|  적   |      |목적지|
 --------------------------------
아이템|      |      |  적   |      |
 --------------------------------
      |      |      |       |      |
 --------------------------------
      |  적   | 포션 |       | USER |
 --------------------------------
 포션 |      |      |       |  적   |
 --------------------------------
```



5. 결과 및 결론

1) 프로젝트 결과

지금까지 배운 내용을 이용하여 mud game을 만들었다.

2) 느낀점

diaplayMap() 함수나 checkXY() 함수 등 구현한 사용자 정의 함수를 User 클래스의 메소드에 포함시킬까 했다. main() 함수에서 이 함수들을 호출해 User 클래스의 멤버변수를 인자로 전달할 때 멤버변수의 직접 접근을 막아놓았기 때문에 메소드를 통해 반환된 값을 전달하는게 복잡하다고 느꼈기 때문이다. 하지만 User 클래스는 유저 정보를 관리하는 역할로 구현했기 때문에 맵의 상황을 체크하는 것은 User 클래스의 역할이 아니라고 생각해서 포함시키지 않았다.

checkState() 함수가 checkGoal() 함수의 역할까지 수행하게 해서 코드를 더 간결하게 할 수 있지만 게임의 목표인 목적지 도달과 목표가 아닌 이벤트 발생을 분리해 놓는 것이 코드의 유지, 보수 측면에서 더 좋은 것 같아서 두 함수를 병합하지 않았다.