전남대학교
CHONNAM NATIONAL UNIVERSITY

[MUD 게임 개발 보고서]

.

■과 목 명 : C++프로그래밍및실습

■ 담 당 교 수 : 김 미 수

■제 출 일 : 2023.10.27

■학 과 : 전기공학과

■학 번 : 200244

■성 명 : 김 민 호

서론

1. 프로젝트 목적 및 배경

7 주차까지 배운 내용에 대한 실습을 위해 진행

2. 목표

간단한 mud 게임 구현

요구사항

1. 사용자 요구 사항

5 x 5 보드판

말을 상하좌우로 움직여 무사히 목적지에 도착하기

2. 기능 계획

기본 기능

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

② 입력 받은 명령에 따라 명령 실행

A. 말 이동 : 상/하/좌/우 입력 시 해당 방향으로 이동 후 지도, 말의 위치 출력

%지도 밖으로 나가게 되면 에러 메시지 출력%

B. 지도 출력 : “지도” 입력 시 현재 지도, 말의 위치 출력

C. 종료 : “종료” 입력 시 프로그램 종료

D. 위 명령 이외에 다른 단어 입력 시 에러 메시지 출력 & 입력 요청 재실행

③ 목적지에 도착하면 “성공”을 출력하고 종료

추가 기능 : hp 상태 조건

① 초기 hp : 20

② 말 1 칸 이동 시 hp 1 감소

③ 지도 출력 시 현재 hp 상태와 함께

④ 다음과 같은 상태일 때 hp 변동

A. 아이템 칸에 위치할 경우 메시지 출력

B. 적 칸에 위치할 경우 hp 2 감소 , 메시지 출력

C. 포션 칸에 위치할 경우 hp 2 증가

⑤ hp 가 0 이 되면 “실패” 출력 후 종료

3. 함수 계획

main 함수: 사용자에게 값을 계속 입력받고, 그에 대한 함수 호출

displayMap() : 지도와 현재 말의 위치 출력 함수

checkNmoveXY() : 이동할 위치가 정상적이면 말을 이동시키는 함수

checkGoal() : 목적지 도착 체크 함수

checkState() : hp 증감 처리, hp 가 0 이 되었는지 체크하고 메시지를 출력하는 함수

설계 및 구현

1. 기능별 구현 사항

전처리 문장

```
#include <iostream>
#include <string>
using namespace std;

const int mapX = 5;
const int mapY = 5;
int user_hp = 20;
// 사용자 정의 함수
bool checkNmoveXY(int& user_x, int mapX, int& user_y, int mapY, int movement_x, int movement_y);
void displayMap(int map[][mapX], int user_x, int user_y);
bool checkGoal(int map[][mapX], int user_x, int user_y);
bool checkState(int map[][mapX], int user_x, int user_y);
```

mapX : 지도의 칸(열) 수

mapY : 지도의 줄(행) 수

user_hp :말의 현재 체력

사용자 정의 함수

checkNmoveXY() : 이동할 위치가 정상적이면 말을 이동시킨 후 이동량에 따라 hp 를 감소시키는 함수

```
// 이동하려는 곳이 유효한 좌표인지 체크하는 함수
bool checkNmoveXY(int& user_x, int mapX, int& user_y, int mapY, int movement_x, int movement_y) {
    bool checkFlag = false;
    user_x += movement_x;
    user_y += movement_y;
    if (user_x >= 0 && user_x < mapX && user_y >= 0 && user_y < mapY) {
        checkFlag = true;
        user_hp -= movement_x + movement_y;
    }
    else {
        cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
        user_x -= movement_x;
        user_y -= movement_y;
    }

    return checkFlag;
}
```

입력 : 지도의 행 ,열 수, 말의 위치(x,y), 이동에 대한 변수

결과 :

맵을 벗어날 경우 메시지 출력

맵을 벗어나지 않을 경우 말 이동

설명 :

말의 위치 (user_x, y)를 대입 연산을 통해 이동 변수(movement_x, y) 만큼 이동시킨다.

If 문을 이용하여 이동한 말의 위치가 맵 안에 존재하면 이동한 만큼 hp 를 감소시키고, checkFlag 를 true 로 설정한다. (이때 checkFlag 는 이동하려는 곳이 유효한 좌표인지 체크하는 변수이다.)

이동한 말의 위치가 맵 밖으로 벗어나면 문장을 출력 후 말의 위치를 대입 연산을 통해 복귀시킨다.

이후 checkFlag 를 반환한다.

displayMap() : 지도와 현재 말의 위치 출력 함수

```cpp
// 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;
    }
}
```

입력 : 지도 정보 배열, 말의 위치(user_x, y)

결과 :지도와 말의 위치를 표 형태로 출력

설명 :

각 칸의 세로줄은 오른쪽에 |를 입력하여 구성한다.

posState 는 지도의 정보를 받는 변수이다.

지도 내 정보에서 0 는 빈칸, 1 은 아이템, 2 는 적, 3 은 포션, 4 는 목적지를 나타낸다. 위 정보를 switch 문을 통해 출력한다.

한 줄을 출력한 후 “----------“를 출력하여 아랫줄을 구성한다.

위의 실행문을 각 층에 대하여 반복한다.

checkGoal() : 목적지 도착 체크 함수

```cpp
// 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;
}
```

입력 : 지도 정보 배열, 말의 위치(user_x, y)

결과 : 목적지 도착 여부 반환

설명 : 지도 내 정보에서 4 는 목적지를 나타낸다.

말의 위치의 지도 내 정보가 4 라면 목적지에 도착한 것이므로 true 값을 반환하고, 아니라면 false 를 반환한다.

checkState() : hp 증감 처리, hp 가 0 이 되었는지 체크하고 메시지를 출력하는 함수

```cpp
bool checkState(int map[][mapX], int user_x, int user_y) {
    if (map[user_y][user_x] == 1) {
        cout << "아이템이 있습니다." << endl;
    }
    else if (map[user_y][user_x] == 2) {
        cout << "적이 있습니다. HP가 2 줄어듭니다." << endl;
        user_hp -= 2;
    }
    else if (map[user_y][user_x] == 3) {
        cout << "포션이 있습니다. HP가 2 늘어납니다." << endl;
        user_hp += 2;
    }

    if (user_hp <= 0) {
        cout << "HP가 0 이하가 되었습니다.실패했습니다.\n게임을 종료합니다.";
        return false;
    }
        return true;
}
```

입력 : 지도 정보 배열, 말의 위치(user_x, y)

결과 : hp 증감 처리, hp 가 0 이하로 내려갔는지 여부 반환, 0 이하로 내려갔다면 메시지 출력

설명 :

말의 위치의 지도 내 정보를 바탕으로 if 문을 사용하여 각 조건에 따른 hp 를 증감 처리, 메시시를 출력한다.

이후 user_hp 가 0 이하로 내려갔는지 if 문으로 확인하여, 0 이하일 경우 메시지 출력 후 fale 값을 반환한다.

Main 함수

변수 초기화 : map : 지도 내 각 칸에 대한 정보, user_x, y : 말의 위치 정보

```
// 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
```

사용자에게 “상”, ”하”, ”좌”, ”우”, “지도”, “종료” 중 하나를 입력 받기

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

cout << "현재 Hp: "<<user_hp << "명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : ";
cin >> user_input;
```

입력 : 명령어

결과 : 현재 hp 상태, 명령어 입력 요청 문장 출력, 명령어 저장

설명 : 현재 hp 상태와 명령어 입력 요청 문장을 출력 후 입력받은 명령어를 user_input 에 저장한다.

입력 받은 명령에 따라 명령 실행

A. 말 이동 : 상/하/좌/우 입력 시 해당 방향으로 이동 후 지도, 말의 위치 출력

%지도 밖으로 나가게 되면 에러 메시지 출력%

```
if (user_input == "상") {
    // 위로 한 칸 올라가기
    bool inMap = checkNmoveXY(user_x, mapX, user_y, mapY, 0, -1);
    if (inMap == true) {
        cout << "위로 한 칸 올라갑니다." << endl;
        displayMap(map, user_x, user_y);
    }
    else
        continue;
}
else if (user_input == "하") {
    // 아래로 한 칸 내려가기
    bool inMap = checkNmoveXY(user_x, mapX, user_y, mapY, 0, 1);
    if (inMap == true) {
        cout << "위로 한 칸 내려갑니다." << endl;
        displayMap(map, user_x, user_y);
    }
    else
        continue;
}
else if (user_input == "좌") {
    // 왼쪽으로 이동하기
    bool inMap = checkNmoveXY(user_x, mapX, user_y, mapY, -1, 0);
    if (inMap == true) {
        cout << "왼쪽으로 이동합니다." << endl;
        displayMap(map, user_x, user_y);
    }
    else
        continue;
}
else if (user_input == "우") {
    // 오른쪽으로 이동하기
    bool inMap = checkNmoveXY(user_x, mapX, user_y, mapY, 1, 0);
    if (inMap == true) {
        cout << "오른쪽으로 이동합니다." << endl;
        displayMap(map, user_x, user_y);
    }
    else
        continue;
```

입력 : 명령어, 지도의 줄, 칸 수, 말의 위치, 지도 내 정보

결과 : 이동 명령에 따라 말 이동, 지도와 말의 위치 출력

설명 :

이동 명령에 따라 checkNmoveXY 함수에 이동 명령에 해당하는 이동 변수를 투입한다.

함수는 말을 이동시켜 말이 지도 밖으로 벗어나면 에러 메시지와 함께 continue 를 따라 main 함수에서 가장 큰 for 문을 반복하여 입력 요청 문장을 재실행하게 된다.

말이 지도 안에 있다면 이동에 대한 문장을 출력한 후, displayMap 함수를 통해 지도와 말의 위치를 출력한다.

B. 지도 출력 : “지도” 입력 시 현재 지도, 말의 위치 출력

C. 종료 : “종료” 입력 시 프로그램 종료

D. 위 명령 이외에 다른 단어 입력 시 에러 메시지 출력 & 입력 요청 재실행

```
else if (user_input == "지도") {
    // 지도 출력 함수 호출
    displayMap(map, user_x, user_y);
}
else if (user_input == "종료") {
    cout << "종료합니다.";
    break;
}
else {
    cout << "잘못된 입력입니다." << endl;
    continue;
}
```

입력 : 명령어, 말의 위치, 지도 내 정보

결과 : 명령 실행, 잘못된 입력에 대한 문장 출력 후 for 문 반복

설명 :

If 문을 사용하여 명령어가 “지도”라면 displayMap 함수를 통해 현재 지도와 말의 위치를 출력한다.

명령어가 “종료”라면 종료한다는 문장을 출력 후, 프로그램을 종료한다.

명령어가 잘못되었다면 에러 에시지를 출력 후 for 문을 반복한다.

목적지에 도착하면 “성공”을 출력하고 종료

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

입력 : 말의 위치, 지도 내 정보

결과 : 목적지 도달 시 문장 출력 후 프로그램 종료

설명 :

checkGoal 함수를 사용하여 말이 목적지에 도달하였는지를 finish 에 반환받는다. If 문을 사용하여 목적지에 도달하였다면 도달 메시지를 출력한 후 종료 메시지와 함께 프로그램을 종료한다.

hp 증감 처리, hp 가 0 이하가 되면 프로그램 종료

```
// 특이사항(hp 변동) 발생 여부 체크, hp가 0이 되면 종료
if (checkState(map, user_x, user_y) == false) {
    break;
}
```

입력 : 말의 위치, 지도 내 정보

결과 : hp 증감 처리, hp 가 0 이하가 되면 종료 문장과 함께 프로그램 종료

설명 :

checkState 를 통해 말의 위치에 따른 hp 의 증감 처리를 한 후 hp 가 0 이하 되었는지를 반환받는다. If 문을 사용하여 값이 false 이면 hp 가 되었다는 것이므로 프로그램을 종료한다.

테스트

1. 기능별 테스트 결과

사용자에게 “상”, ”하”, ”좌”, ”우”, “지도”, “종료” 중 하나를 입력 받기

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

말 이동 : 상/하/좌/우 입력 시 해당 방향으로 이동 후 지도, 말의 위치 출력

```
현재 Hp: 13명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : 상
위로 한 칸 올라갑니다.
     |아이템|  적  |     |목적지|
 -------------------------------
아이템|     |     |  적  |     |
 -------------------------------
     |     | USER |     |     |
 -------------------------------
     |  적  | 포션 |     |     |
 -------------------------------
 포션 |     |     |     |  적  |
 -------------------------------
현재 Hp: 12명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : |
```

```
현재 Hp: 19명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : 하
위로 한 칸 내려갑니다.
     |아이템|  적  |     |목적지|
 -------------------------------
아이템| USER |     |  적  |     |
 -------------------------------
     |     |     |     |     |
 -------------------------------
     |  적  | 포션 |     |     |
 -------------------------------
 포션 |     |     |     |  적  |
 -------------------------------
현재 Hp: 18명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : |
```

```
현재 Hp: 18명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) :좌
왼쪽으로 이동합니다.
     |아이템|  적  |     |목적지|
 -------------------------------
USER |     |     |  적  |     |
 -------------------------------
     |     |     |     |     |
 -------------------------------
     |  적  | 포션 |     |     |
 -------------------------------
 포션 |     |     |     |  적  |
 -------------------------------
아이템이 있습니다.

현재 Hp: 17명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : |
```

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

현재 Hp: 19명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : |
```

%지도 밖으로 나가게 되면 에러 메시지 출력%

```
현재 Hp: 19명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : 우
오른쪽으로 이동합니다.
     |아이템| USER |     |목적지|
 -------------------------------
아이템|     |     |  적  |     |
 -------------------------------
     |     |     |     |     |
 -------------------------------
     |  적  | 포션 |     |     |
 -------------------------------
 포션 |     |     |     |  적  |
 -------------------------------
적이 있습니다. HP가 2 줄어듭니다.

현재 Hp: 16명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : 상
맵을 벗어났습니다. 다시 돌아갑니다.

현재 Hp: 16명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : |
```

지도 출력 : “지도” 입력 시 hp 상태와 함께 현재 지도, 말의 위치 출력

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

위 명령 이외에 다른 단어 입력 시 에러 메시지 출력 & 입력 요청 재실행

```
현재 Hp: 20명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : dn
잘못된 입력입니다.
현재 Hp: 20명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : |
```

말 1 칸 이동 시 hp 1 감소

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

현재 Hp: 19명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : |
```

아이템 칸에 위치할 경우 메시지 출력

```
현재 Hp: 18명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) :좌
왼쪽으로 이동합니다.
     |아이템|  적  |     |목적지|
 -------------------------------
 USER |     |     |  적  |     |
 -------------------------------
     |     |     |     |     |
 -------------------------------
     |  적  | 포션 |     |     |
 -------------------------------
 포션 |     |     |     |  적  |
 -------------------------------
아이템이 있습니다.

현재 Hp: 17명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) :
```

적 칸에 위치할 경우 hp 2 감소 , 메시지 출력

```
 -------------------------------
현재 Hp: 15명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : 하
위로 한 칸 내려갑니다.
     |아이템|  적  |     |목적지|
 -------------------------------
아이템|     |     |  적  |     |
 -------------------------------
     |     |     |     |     |
 -------------------------------
     | USER | 포션 |     |     |
 -------------------------------
 포션 |     |     |     |  적  |
 -------------------------------
적이 있습니다. HP가 2 줄어듭니다.

현재 Hp: 12명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) :
```

포션 칸에 위치할 경우 hp 2 증가

```
현재 Hp: 12명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : 우
오른쪽으로 이동합니다.
     |아이템|  적  |     |목적지|
 -------------------------------
아이템|     |     |  적  |     |
 -------------------------------
     |     |     |     |     |
 -------------------------------
     |  적  | USER |     |     |
 -------------------------------
 포션 |     |     |     |  적  |
 -------------------------------
포션이 있습니다. HP가 2 늘어납니다.

현재 Hp: 13명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) :
```

2. 최종 테스트 스크린샷

① 명령어로 “종료”를 입력하였을 경우

```
현재 Hp: 20명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : 종료
종료합니다.
C:\Users\200244\코딩 파일\Visual Studio 2202\C++\mudgame\x64\Debug\mudgame.exe(프로세스 17632개)이(가) 종료되었습니다(코
드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

② Hp 가 0 이하로 내려갔을 경우

```
현재 Hp: 4명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : 좌
왼쪽으로 이동합니다.
     | USER |  적  |     |목적지|
 -------------------------------
아이템|     |     |  적  |     |
 -------------------------------
     |     |     |     |     |
 -------------------------------
     |  적  | 포션 |     |     |
 -------------------------------
 포션 |     |     |     |  적  |
 -------------------------------
아이템이 있습니다.
현재 Hp: 3명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : 우
오른쪽으로 이동합니다.
     |아이템| USER |     |목적지|
 -------------------------------
아이템|     |     |  적  |     |
 -------------------------------
     |     |     |     |     |
 -------------------------------
     |  적  | 포션 |     |     |
 -------------------------------
 포션 |     |     |     |  적  |
 -------------------------------
적이 있습니다. HP가 2 줄어듭니다.
HP가 0 이하가 되었습니다.실패했습니다.
게임을 종료합니다.
```

③ 말이 목적지에 도달하였을 경우

```
현재 Hp: 16명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : 우
오른쪽으로 이동합니다.
     |아이템|  적  | USER |목적지|
 -------------------------------
아이템|     |     |  적  |     |
 -------------------------------
     |     |     |     |     |
 -------------------------------
     |  적  | 포션 |     |     |
 -------------------------------
 포션 |     |     |     |  적  |
 -------------------------------
현재 Hp: 15명령어를 입력하세요(상, 하, 좌, 우, 지도, 종료) : 우
오른쪽으로 이동합니다.
     |아이템|  적  |     | USER |
 -------------------------------
아이템|     |     |  적  |     |
 -------------------------------
     |     |     |     |     |
 -------------------------------
     |  적  | 포션 |     |     |
 -------------------------------
 포션 |     |     |     |  적  |
 -------------------------------
목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.
```

결과 및 결론

1. 프로젝트 결과

1. 기초적인 게임 제작을 체험할 수 있었습니다.
2. 명령어 선택지를 입력받아 입력에 따라 명령을 수행하는 프로그램을 구현하였습니다.
3. 예외적인 경우(맵 밖을 벗어나는 버그가 될 수 있는 부분)들을 다루었습니다.
4. 플레이어의 행동과 환경에 따라 플레이어의 스탯(hp)를 처리하는 코드를 작성하였습니다.

게임 개발에서 기초라고 할 수 있는 부분들을 간단하게 구혔해보았습니다.

2. 느낀 점

실습과 프로젝트를 통해 프로그래밍으로 개발할 수 있는 다양한 것들을 체험할 수 있었습니다. 자료를 관리하는 프로그램, 데이터 계산 프로그램, 게임 개발 프로젝트, 추천 프로그램 등을 진행하면서 점점 프로그래밍에 흥미를 느끼게 되었습니다. 교수님께서도 열정적이셔도 저도 더 배우고 싶어집니다. 다만 조금 아쉬운 점은 과제를 준비할 때에나 수업 중에 끝내지 못한 실습을 집에서 다시 해보는 데 필요한 시간이 부족하다고 느낍니다. 더 나은 퀄리티를 위해서라도 제출 기한이 하루 정도 늦춰지면 좋겠습니다.