C++프로그래밍 및 실습

Mud game

보고서

제출일자: 2024/11/03
제출자명: 손유채
제출자 학번: 214940

1. 서론

1) 프로젝트 목적 및 배경

Mud 게임을 알아보고 1주차부터 7주차 까지의 내용으로 c++ 수업 내용 실습을 위해 프로젝트 진행

2) 프로젝트 목표

Mud 게임의 의미와 Mud게임을 c++ 프로그래밍으로 구현

2. 요구 사항

1) 사용자의 요구사항

상하좌우로 이동하며 무사히 목적지에 도착하기

2) 기능 계획

1. 플레이어의 체력을 20, 이동시 1의 체력이 감소하고 체력이 0이 되기 전에 목적지에 도달한다.

1-1. 전체 지도에는 적, 아이템, 물약이 있으며 이와 상호작용할 시 그에 따른 효과가 적용된다.

2. 사용자에게 상, 하, 좌, 우, 지도, 종료 중 하나를 입력 받는다.

2-1. 상/하/좌/우 입력 시 해당 방향으로 이동 후 지도를 출력한다

2-2. 지도 입력 시 전체지도가 출력되며 현재 위치를 알 수 있게 된다.

2-3. 다른 것을 입력하면 에러메시지 출력 후 재 입력을 요청한다

3. 지도 밖으로 나가면 에러메시지가 출력되게 한다.

4. 목적지에 도착 시 성공하며 메시지 출력 후 종료된다.

3) 함수 계획

1. 플레이어의 체력을 전역변수로 지정하여 다른 함수에서도 쉽게 접근이 되게 한다.

2. main 함수에 cin으로 입력을 받는다

2-1. 상/하/좌/우 입력에 대한 함수를 만들어 불필요한 코드의 중복을 줄인다.

3. 설계 및 구현

1) 기능별 구현 사항

```
// 플레이어 움직임(상하좌우) 간결화 함수
//포인터를 사용해 함수 밖의 좌표까지 영향이 미치게함
void movePlayer(int &user_x, int &user_y, int ax, int ay, int map[][mapX]) {
    int new_x = user_x + ax;
    int new_y = user_y + ay;

    if (checkXY(new_x, mapX, new_y, mapY) == false) {
        cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
    } else {
        user_x = new_x;
        user_y = new_y;
        health -= 1; // 정상 이동 시 HP 1 감소
        if (ay == -1) cout << "위로 한 칸 올라갑니다." << endl;
        else if (ay == 1) cout << "아래로 한 칸 내려갑니다." << endl;
        else if (ax == -1) cout << "왼쪽으로 한 칸 이동합니다." << endl;
        else if (ax == 1) cout << "오른쪽으로 한 칸 이동합니다." << endl;
        displayMap(map, user_x, user_y);
    }
}
```

입력 - int &user_x = 유저 x값 pointer, int &user_y = 유저 y값 pointer

(int ax , int ay) = 유저 입력에 따른 변화 될 좌표 값

int map[][] = 전체 지도

반환값 – 없음

결과 - 지도를 벗어나면 에러메시지 출력

지도 안에 좌표가 존재하면 전체지도 출력과 이동 메시지 출력

설명 - 변화좌표 값 계산 후 지도를 벗어나면 에러메시지 출력

지도 안에 변화좌표가 존재하면 적용 후 메시지 출력 및 지도 출력

```
// 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;
}
```

입력 - int user_x = 유저 x값, int mapX = 지도 크기의 x값(가로)

int user_y = 유저 y값, int mapY= 지도 크기의 y값(세로)

반환값 – True or False

결과 – 이동한 좌표가 유효좌표 여부에 따른 T/F반환

설명 – 유저 좌표와 지도 크기를 비교해 T/F반환

```
// 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 - int map[][] = 전체 지도, int user_x = 유저 x값, int user_y = 유저 y값

반환값 – True or False

결과 – 이동한 좌표가 목적지 여부에 따른 T/F반환

설명 – 유저 좌표와 목적지 좌표를 비교 후 T/F반환

```
// 유저의 위치를 확인하는 함수
void checkState(int map[][mapX], int user_x, int user_y)
{
    if (map[user_y][user_x] == 1)
    {
        cout << "아이템이 있습니다." << endl;
    }
    if (map[user_y][user_x] == 2)
    {
        cout << "적이 있습니다. HP가 2 줄어듭니다" << endl;
        health -= 2; // 적과 조우 후 HP 2감소
    }
    if (map[user_y][user_x] == 3)
    {
        cout << "포션이 있습니다. HP가 2 증가합니다" << endl;
        health += 2; // 포션 얻은 후 HP 2증가
    }
}
```

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

반환값 – 없음

결과 – 현재 유저 좌표에 있는 아이템/적/포션의 여부에 따른 메시지 출력

설명 – 유저 좌표와 지도 좌표를 비교 후 메시지 출력

```
// 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 - int map[][] = 전체 지도, int user_x = 유저 x값, int user_y = 유저 y값

반환값 – 없음

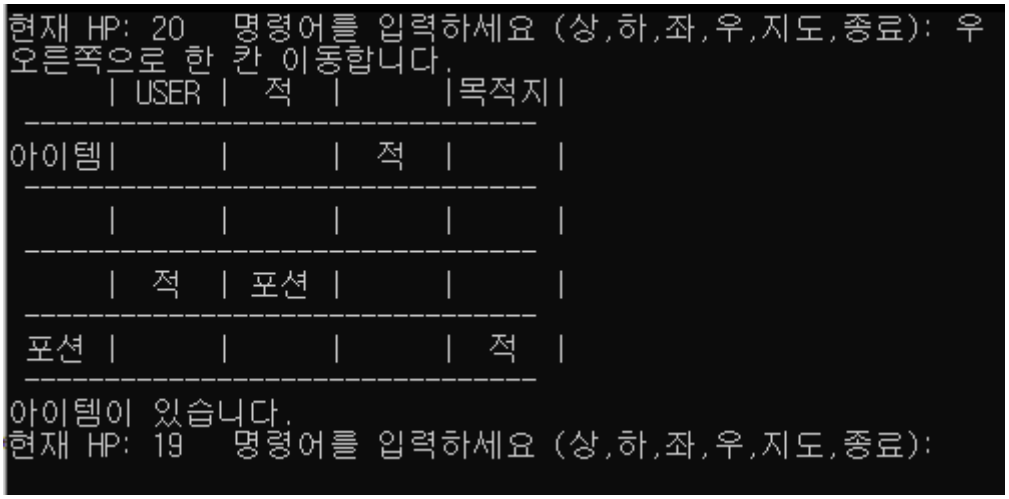
결과 – 전체지도 출력 및 사용자 위치 출력

설명 – 2차원 배열의 지도를 출력하다 특정 사물, 플레이어가 있는 좌표를 받을 시 정보를 출력

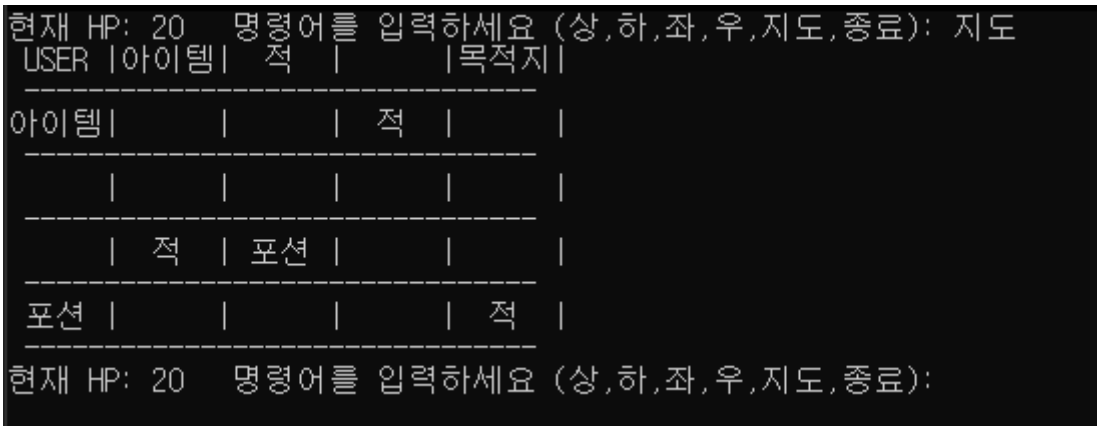
4. 테스트

1) 기능별 테스트 결과

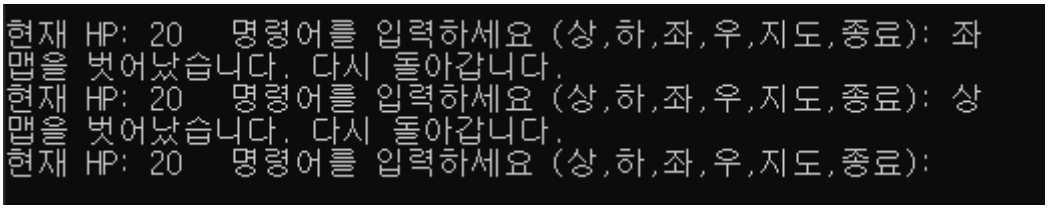
1. 시작 시 플레이어의 체력을 20, 상하좌우 입력 받고 이동 후 지도 출력



2. 지도 입력 시 지도 출력



3. 지도 밖으로 나갈 시 에러메시지 출력



4. 다른 명령어 입력 시 에러메시지 출력

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

5. 목적지 도착 시 게임 종료

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

1) 최종 테스트 스크린샷

```
현재 HP: 20   명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 한 칸 이동합니다.
      | USER |  적  |      |목적지|
 ---------------------------------
아이템|      |      |  적  |      |
 ---------------------------------
      |      |      |      |      |
 ---------------------------------
      |  적  | 포션 |      |      |
 ---------------------------------
 포션 |      |      |      |  적  |
 ---------------------------------
아이템이 있습니다.
현재 HP: 19   명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 한 칸 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): 우
오른쪽으로 한 칸 이동합니다.
      |아이템|  적  | USER |목적지|
 ---------------------------------
아이템|      |      |  적  |      |
 ---------------------------------
      |      |      |      |      |
 ---------------------------------
      |  적  | 포션 |      |      |
 ---------------------------------
 포션 |      |      |      |  적  |
 ---------------------------------
현재 HP: 15   명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 한 칸 이동합니다.
      |아이템|  적  |      | USER |
 ---------------------------------
아이템|      |      |  적  |      |
 ---------------------------------
      |      |      |      |      |
 ---------------------------------
      |  적  | 포션 |      |      |
 ---------------------------------
 포션 |      |      |      |  적  |
 ---------------------------------
목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.
```

5. 결과 및 결론

1) 프로젝트 결과

Mud 게임을 만들었습니다.

1) 느낀 점

Mud game을 만들면서 함수를 다시 한번 복습하는 좋은 계기가 되었고 예전부터 게임에 관심이 많았었기에 흥미롭게 만들었습니다. 기간이 촉박하여 주어진 베이스 코드를 주로 이용하게 되었지만 제 스스로 만들기회가 오면 좋겠습니다.