

# 1.서론

프로젝트 배경: 수업시간에 배운 사용자 정의함수를 보다 익숙하기 위함. main함수에 직접 기능을 입력하는 것 보다 함수를 정의하여서 main함수에서 가져다가 쓰는 게 훨씬 수월하게 작업을 할 수 있기 때문

프로젝트 내용: 간단한 게임으로 사용자에게 입력을 받고 각각에 맵에는 아이템, 적,포션등으로 상대방의 hp를 변화시키는 효과가 있음 상대방은 목적지로 hp가 사라기전에 이동해야함

## 2.요구사항

- 1.체력 20으로 시작
- 2.이동시 체력 -1
- 3.hp상태 출력
- 4.hp=0이면 게임 종료
- 5.무기/포션/아이템등을 만났을 때 메시지 출력
- 6.함수로 간략화

## 3.설계 및 구현

```

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

const int mapX = 5;
const int mapY = 5;
int hp = 20;

// 사용자 정의 함수
bool checkXY(int user_x, int mapX, int user_y, int mapY);
void displayMap(int map[][mapX], int user_x, int user_y);
bool checkGoal(int map[][mapX], int user_x, int user_y);
int MoveMinus(int&x);
void checkState(int map[][mapX], int user_x, int user_y);

```

-입력: 따로 없음

Hp=20을 전역변수로 선언을 하였기 때문에 아래에 있는 main 함수와 사용자 정의 함수에서도 사용이 가능함

```

//유저 이동시마다 hp 감소시키는 함수
int MoveMinus(int&x) {
    if (true)
        x-= 1;
    return x;
}

```

-입력:int&x 사용하여 원형 x값도 변화되게함

반환값:x-1을 하고 main함수로 다시 반환을 함

결과:MoveMinus함수를 호출할때마다 대입된 변수는 -1이 됨

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

```
if (user_input == "상") {
    // 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 {
        // 맵이 벗어 나지 않으므로 실행 moveminus함수로 -1를 시켜줌
        checkState(map, user_x, user_y);
        cout << "현재 HP: "<<MoveMinus(hp);
        cout << " 위로 한 칸 올라갑니다." << endl;
        displayMap(map, user_x, user_y);
        //체력이 0일경우 종료
        if (hp <= 0)
        {
            cout << "HP가 0 이하가 되었습니다. 실패했습니다.";
            cout << "게임을 종료합니다.";
            break;
        }
    }
}
```

입력:

- checkXY:유저가 어떤 상태인지 확인해주고 그 상황에 맞게 유저의 HP조정과 상황을 출력해줌

-inMap: 맵의 크기

User\_x: 유저의 x값

User\_Y: 유저의 y값

cnckState:유저의 위치를 나타냄

MoveMinus:유저 이동시 hp -1 감소

displayMap:지도 보여줌

반환값:없음

결과:현재 Hp상태와 유저의 현재 위치를 표시시킴, 만약 hp가 0이하가 되면 게임을 종료 시킴

```

//유저가 어떤 상태인지에 따라 반환값을 정하는 함수
void checkState(int map[][mapX], int user_x, int user_y) {
    switch (map[user_y][user_x]) {
        case 1:
            cout << "아이템이 있습니다" << endl;
            break;
        case 2:
            cout << "적이 있습니다. HP가 2 줄어듭니다." << endl;
            userHP -= 2;
            break;
        case 3:
            cout << "포션이 있습니다. HP가 2 늘어납니다." << endl;
            userHP += 2;
            break;
    }
}

```

-입력:

2차원 배열 크기.user\_x,user\_y를 매개변수로 받음

반환값:

유저의 위치에 따른 값에 따라 1인 경우에는 "아이템이 있습니다" 출력

2인 경우 "적이 있습니다. HP가 2 줄어듭니다."를 출력후 hp-2

3인 경우 "포션이 있습니다. HP가 2늘어납니다."를 출력후 hp+2

결과

각각 상황에 따라 HP를 늘리거나 감소 시키고 그에 대한 설명을 출력시킴

```

// 지도와 사용자 위치 출력하는 함수
//배열의 크기를 정의할때 열부분만 적어도 인식을함! 원래는 둘다 적어야하지만!
void displayMap(int map[][mapX], int user_x, int user_y) {
    //행부분을 Y좌표로 설정하는 이유는 기본적으로 Y좌표를 행부분으로 인식하기때문이다!
    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;
    }
}

```

-입력값

2차원 배열 크기.user\_x,user\_y를 매개변수로 받음

-반환값:

-사용자의 현재 위치와 지도를 출력시킴

## 4.테스트

```
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): 하
아이템이 있습니다
현재 HP:19 위로 한 칸 내려갑니다.
|아이템| 적 | 목적지|
```

```
-----
USER |      |      |  적  |      |
-----
|      |      |      |      |
-----
|  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
```

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

-현재 HP를 표시! 사용자가 맵을 나갈라고 할 때 맵을 벗어났다고 출력후 다시  
입력받음,이동시 맵을 보여주고 hp를 1감소 시킴

```
현재 HP:17 명령어를 입력하세요 (상,하,좌,우,지도,종료): 하
포션이 있습니다. HP가 2 늘어납니다.
현재 HP:18 위로 한 칸 내려갑니다.
|아이템| 적 | 목적지|
```

```
-----
아이템|      |      |  적  |      |
-----
|      |      |      |      |
-----
|  적  | 포션 |      |      |
-----
USER |      |      |      |  적  |
-----
```

```
현재 HP:17 명령어를 입력하세요 (상,하,좌,우,지도,종료): 상
적이 있습니다. HP가 2 줄어듭니다.
현재 HP:14 위로 한 칸 올라갑니다.
|아이템| 적 | 목적지|
```

```
-----
아이템|      |      |  적  |      |
-----
|      |      |      |      |
-----
| USER | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
```



-간단한 mud game을 구현해보았다. 이 프로젝트 경험으로 기능을 구현할 때 main 함수로 쭉 길게 작성하는 것보다 함수를 이용하여서 main함수를 최대한 알아보기 쉽게 간략하는 것의 중요성을 알게 되었다. 나중에 프로젝트를 할 때 좋은 경험이 될 것 같다.