

프로젝트 목적 및 배경: 7주차까지 배운 내용에 대한 실습을 위해 진행

목표: 간단한 Mud 게임구현

추가 기능 계획:

- ① 유저는체력20을가지고게임시작
- ② 사용자가이동할때마다사용자체력1씩감소
- ③ 처음 명령문을 입력 받을 때마다 HP 함께 출력
- ④ HP가0이되면"실패"를출력하고종료
- ⑤ 무기/갑옷, 포션, 적을 만났을때 그에 대한 메시지를 출력

함수 계획:

맵을 벗어났는지 체크 하는 코드최적화(함수화)

//한글로 상, 하, 좌, 우, 지도, 종료 입력이 받아
들여지지 않아서 영어로 대체하였음.

- ① 유저는체력20을가지고게임시작

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

const int mapX = 5;
const int mapY = 5;

// 유저 체력
int hp=20;
```

전역변수(checkState에서 hp에 편하게 전근하기 위해)로 hp 선언 후 20으로 초기화.

② 사용자가 이동할 때마다 사용자 체력 1씩감소

```
is_in();
if(inMap==true) {
    cout << "위로 한 칸 올라갑니다." << endl;
    displayMap(map, user_x, user_y);
    hp--;
}
}
else if (user_input == "down") {
    // TODO: 아래로 한 칸 내려가기
    user_y += 1;
    is_in();
    if(inMap==true) {
        cout << "위로 한 칸 내려갑니다." << endl;
        displayMap(map, user_x, user_y);
        hp--;
    }
}

else if (user_input == "left") {
    // TODO: 왼쪽으로 이동하기
    user_x -= 1;
    is_in();
    if(inMap==true) {
        cout << "왼쪽으로 이동합니다." << endl;
        displayMap(map, user_x, user_y);
        hp--;
    }
}

else if (user_input == "right") {
    // TODO: 오른쪽으로 이동하기
    user_x += 1;
    is_in();
    if(inMap==true) {
        cout << "오른쪽으로 이동합니다." << endl;
        displayMap(map, user_x, user_y);
        hp--;
    }
}
}
```

상하좌우 이동할 때 마다 hp--;

③ 처음 명령문을 입력 받을 때마다 HP 함께 출력

```
cout << "현재 HP: "<<hp <<" 명령어를 입력하세요 (up,down,left,right,map,finish): ";
```

명령어 입력하세요 출력 전 hp 출력

④ HP가 0이되면 "실패"를 출력하고 종료

```
//Hp가 0이하로 되었는지 체크
if(hp<=0){
    cout<<"HP가 0이하가 되었습니다. 실패했습니다.";
    return 0;
}

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

목적지에 도달하면서 피가 0이되면 도착했다고 해야 하는지 실패했다고 해야 하는지 모르겠어서 그냥 피가 0되면 실패했다고 출력하도록 목적지 도달 체크 전에 썼음.

⑤ 무기/갑옷, 포션, 적을 만났을때 그에 대한 메시지를 출력

```

//아이템,포션,적 만났을때 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:
            hp-=2;
            cout<<"적이 있습니다. HP가 2 줄어듭니다."<<endl;;
            break;
        case 3:
            hp+=2;
            cout<<"포션이 있습니다. HP가 2 늘어납니다."<<endl;
            break;
    }
}

```

checkState 함수를 작성하고

```

// map와 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;
    }
    checkState(map, user_x, user_y); //호출
}

```

displayMap 맨 마지막에 호출 pdf 예시처럼 맵 아래에 뭘 만났는지 출력된다.

맵을 벗어났는지 체크 하는 코드최적화(함수화)

```
//맵벗어났는지 체크함수
void is_in(){
    inMap = checkXY(user_x, mapX, user_y, mapY);
    if (inMap == false) {
        cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
        if(user_input=="up")
            user_y+=1;
        if(user_input=="down")
            user_y-=1;
        if(user_input=="left")
            user_x+=1;
        if(user_input=="right")
            user_x-=1;
    }
}
```

맵을 벗어났는지 확인하고 입력 상하좌우를 확인해서 전 좌표로 복구한다.

```
is_in();
if(inMap==true) {
    cout << "위로 한 칸 올라갑니다." << endl;
    displayMap(map, user_x, user_y);
    hp--;
}
```

If 였던 위 코드가 바뀌었으니 원래 esle 였던 아래 코드도 제대로 동작하도록 바꿔준다.

1번테스트

현재 HP: 20 명령어를 입력하세요 (up,down,left,right,map,finish):

2번테스트

```

현재 HP: 20 명령어를 입력하세요 (up,down,left,right,map,finish): right
오른쪽으로 이동합니다.
      | USER | 적   |      |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션  |      |      |      |  적  |
-----
아이템이 있습니다.
현재 HP: 19 명령어를 입력하세요 (up,down,left,right,map,finish): down
위로 한 칸 내려갑니다.
      |아이템|  적  |      |목적지|
-----
아이템| USER |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션  |      |      |      |  적  |
-----
현재 HP: 18 명령어를 입력하세요 (up,down,left,right,map,finish):

```

3번테스트=1번테스트와 동일

4번테스트

```

현재 HP: 1 명령어를 입력하세요 (up,down,left,right,map,finish): up
위로 한 칸 올라갑니다.
      |아이템| USER |      |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션  |      |      |      |  적  |
-----
적이 있습니다. HP가 2 줄어듭니다.
HP가 0이하가 되었습니다. 실패했습니다.

```

5번테스트

현재 HP: 20 명령어를 입력하세요 (up,down,left,right,map,finish): right
오른쪽으로 이동합니다.

	USER	적		목적지
아이템			적	
	적	포션		
포션				적

아이템이 있습니다.

현재 HP: 19 명령어를 입력하세요 (up,down,left,right,map,finish): right
오른쪽으로 이동합니다.

	아이템	USER		목적지
아이템			적	
	적	포션		
포션				적

적이 있습니다. HP가 2 줄어듭니다.

현재 HP: 16 명령어를 입력하세요 (up,down,left,right,map,finish):

현재 HP: 14 명령어를 입력하세요 (up,down,left,right,map,finish): down
위로 한 칸 내려갑니다.

	아이템	적		목적지
아이템			적	
	적	USER		
포션				적

포션이 있습니다. HP가 2 늘어납니다.

현재 HP: 15 명령어를 입력하세요 (up,down,left,right,map,finish):

아이템, 포션, 적 모두 정상 작동

목적지 도달시 정상적으로 종료.


```

현재 HP: 11 명령어를 입력하세요 (up,down,left,right,map,finish): up
위로 한 칸 올라갑니다.
  |아이템|  적  |      | USER |
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션  |      |      |      |  적  |
-----
목적지에 도착했습니다! 축하합니다!
게임을 finish합니다.

```

결과 및 결론: 간단한 머드 게임을 만들어보았다 딱히 어려운 것은 없었다.