

#mud_game

223310 주소연

1. 서론

프로젝트 목적 및 배경

: 7주차까지 배운 내용에 대한 실습을 위해 진행

목표

: 간단한 Mud 게임 구현

2. 요구사항

사용자 요구사항

: 유저가 상하좌우로만 이동하며 목적지에 도착하는 게임

기능 계획

: ① 사용자에게 “up”, “down”, “left”, “right”, “map”, “exit” 중 하나를 입력 받기(up/down/left/right 입력시 해당 방향으로 이동 후 지도 출력, “map”를 입력하면 전체 지도와 함께 현재 위치를 출력, 이 중 다른 것을 입력하면 에러 메시지 출력 후 재 입력 요청)

② 지도 밖으로 나가게 되면 에러 메시지 출력

③ 유저는 체력 20을 가지고 게임 시작

④ 사용자가 이동할 때 마다 사용자 체력 1씩 감소

⑤ 처음 명령문을 입력 받을 때 마다 HP 함께 출력

⑥ HP가 0이 되면 “실패”를 출력하고 종료

⑦ 무기/갑옷, 포션, 적을 만났을 때 그에 대한 메시지를 출력

⑧ 목적지에 도착하면 “성공”을 출력하고 종료

함수 계획

: ① 메인 함수: 사용자에게 값을 계속 입력받고, 그에 대한 함수 호출

② 지도와 현재 위치 출력 함수: displayMap()

③ 사용자 위치 체크 함수: checkXY()

④ 목적지에 도착 체크 함수: checkGoal()

⑤ 상태 체크 함수: checkState()

3. 설계 및 구현

기능 별 구현 사항

① 사용자에게 “up”, “down”, “left”, “right”, “map”, “exit” 중 하나를 입력 받기(up/down/left/right 입력시 해당 방향으로 이동 후 지도 출력, “map”를 입력하면 전체 지도와 함께 현재 위치를 출력, 이 중 다른 것을 입력하면 에러 메시지 출력 후 재 입력 요청) + ② 지도 밖으로 나가게 되면 에러 메시지 출력 + ③ 사용자가 이동할 때 마다 사용자 체력 1씩 감소 + ④ 처음 명령문을 입력 받을 때 마다 HP 함께 출력

```
// 게임 시작
while (1)
{
    // 사용자에게 계속 입력받기 위해 무한 루프
    cout << "현재 체력: " << HP; // 현재체력출력
    // 사용자의 입력을 저장할 변수
    string user_input = "";

    cout << " 명령어를 입력하세요 (up,down,left,right,map,exit): ";
    // 상,하,좌,우,지도,종료를 각각 up,downright,left,map, exit로 바꿨을 한글인식불가
    cin >> user_input;

    if (user_input == "up")
    {
        // 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
        {
            cout << "위로 한 칸 올라갑니다." << endl;
            HP -= 1; // 이동시 hp감소
            displayMap(map, user_x, user_y);
            checkState(map, user_x, user_y, HP); // 현재 상태 체크
        }
    }
    else if (user_input == "down")
    {
        // TODO: 아래로 한 칸 내려가기
        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
        {
            cout << "위로 한 칸 내려갑니다." << endl;
            HP -= 1; // 이동시 hp감소
            displayMap(map, user_x, user_y);
            checkState(map, user_x, user_y, HP); // 현재 상태 체크
        }
    }
}
```

```

else if (user_input == "left")
{
    // TODO: 왼쪽으로 이동하기
    user_x -= 1;
    bool inMap = checkXY(user_x, mapX, user_y, mapY);

    if (inMap == false)
    {
        cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
        user_x += 1;
    }
    else
    {
        cout << "왼쪽으로 이동합니다." << endl;
        HP -= 1; // 여기서 hp 감소
        displayMap(map, user_x, user_y);
        checkState(map, user_x, user_y, HP); // 현재 상태 체크
    }
}
else if (user_input == "right")
{
    // TODO: 오른쪽으로 이동하기
    user_x += 1;
    bool inMap = checkXY(user_x, mapX, user_y, mapY);
    if (inMap == false)
    {
        cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
        user_x -= 1;
    }
    else
    {
        cout << "오른쪽으로 이동합니다." << endl;
        HP -= 1; // 여기서 hp 감소
        displayMap(map, user_x, user_y);
        checkState(map, user_x, user_y, HP); // 현재 상태 체크
    }
}
else if (user_input == "map")
{
    // TODO: 지도 보여주기 함수 호출
    displayMap(map, user_x, user_y);
}
else if (user_input == "exit")
{
    cout << "종료합니다.";
    break;
}
else
{
    cout << "잘못된 입력입니다." << endl;
    continue;
}
}

```

- 입력 (블록/함수에 입력되는 변수, 값들과 설명)
- HP: 현재 플레이어의 체력. (초기값: 20)
- user_x: 플레이어의 현재 x 좌표 (가로 위치)
- user_y: 플레이어의 현재 y 좌표 (세로 위치)
- user_input: 사용자로부터 입력받은 명령어 (문자열)
- up: 위로 이동하는 명령어
- down: 아래로 이동하는 명령어
- left: 왼쪽으로 이동하는 명령어
- right: 오른쪽으로 이동하는 명령어
- map: 현재 맵을 보여주는 명령어

exit: 게임을 종료하는 명령어

- 반환값 (함수의 경우 작성)

함수 호출 결과:

checkXY: 유효한 좌표인지 확인하여 true 또는 false 반환

displayMap: 맵을 출력하고 결과적으로 사용자 위치를 표시

checkState: 현재 위치의 상태를 확인하고 해당 메시지를 출력

- 결과 (블록/함수가 종료된 결과)

사용자가 유효한 명령어를 입력하면 플레이어의 위치가 업데이트되고 HP가 1 감소된다

플레이어가 적을 만날 경우 HP가 추가로 2 감소된다. 상태가 체크되어 적이나 아이템 등의 메시지가 출력된다. 사용자가 map을 입력하면 현재 상태의 맵이 출력된다. 사용자가 exit를 입력하면 "종료합니다."라는 메시지를 출력하고 게임이 종료된다. 잘못된 입력을 받을 경우 "잘못된 입력입니다."라는 메시지를 출력한다.

- 설명 (코드 내 작동 순서, 내용 등 추가 설명)

프로그램은 시작할 때 플레이어의 체력을 20으로 설정하고 현재 위치를 (0, 0)으로 초기화한다. 사용자가 명령어를 입력하면, 해당 명령어에 따라 플레이어의 위치가 업데이트되거나 맵이 출력된다 ("up" 입력 시 y 좌표를 감소시켜 위로 이동하고, 유효한 이동인지 checkXY 함수를 통해 확인한다. 이동이 유효하지 않다면 "맵을 벗어났습니다. 다시 돌아갑니다."라는 메시지를 출력하고 원래 위치로 돌아간다. 이동이 유효하면 "위로 한 칸 올라갑니다."라는 메시지를 출력하고 HP를 1 감소시키며 현재 맵을 출력한다. 그 후, checkState 함수를 호출하여 현재 위치의 상태를 체크하고 해당 메시지를 출력한다.)

⑤ HP가 0이 되면 "실패"를 출력하고 종료

```
// 체력이 0이되었는지 체크
if (HP <= 0)
{
    cout << "HP가 0이 되었습니다. 실패입니다!" << endl;
    break;
}
```

- 입력 (블록/함수에 입력되는 변수, 값들과 설명)

HP: 현재 플레이어의 체력

- 결과 (블록/함수가 종료된 결과)

HP가 0 이하일 경우: "HP가 0이 되었습니다. 실패입니다!"라는 메시지를 출력한다.

break 문으로 인해 게임 루프가 종료되고, 프로그램의 실행이 끝난다.

⑥ 무기/갑옷, 포션, 적을 만났을 때 그에 대한 메시지를 출력

```
// 상태 체크 함수
void checkState(int map[][mapX], int user_x, int user_y, int &HP)
{
    int posState = map[user_y][user_x]; // 현재 위치의 상태를 확인
    if (posState == 2) // 적 만났을 경우
    {
        cout << "적이 있습니다 HP가 2 줄어듭니다." << endl;
        HP -= 2; // 적과의 전투로 HP 감소
    }
    else if (posState == 1) // 아이템 만났을 경우
    {
        cout << "아이템이 있습니다" << endl;
    }
    else if (posState == 3) // 포션 만났을 경우
    {
        cout << "포션이 있습니다 HP가 2 증가합니다." << endl;
        HP += 2; // 포션으로 HP 증가
    }
}
```

- 입력 (블록/함수에 입력되는 변수, 값들과 설명)

map[][mapX]: 2차원 배열로, 게임의 맵 상태를 나타낸다

user_x: 플레이어의 현재 x좌표

user_y: 플레이어의 현재 y좌표

HP: 플레이어의 체력

- 반환값 (함수의 경우 작성)

이 함수는 반환값이 없다 (void 타입)

- 결과 (블록/함수가 종료된 결과)

플레이어가 적을 만났을 경우: "적이 있습니다 HP가 2 줄어듭니다."라는 메시지를 출력하고, HP가 2 감소한다.

플레이어가 아이템을 만났을 경우: "아이템이 있습니다"라는 메시지를 출력합니다.

플레이어가 포션을 만났을 경우: "포션이 있습니다 HP가 2 증가합니다."라는 메시지를 출력하고, HP가 2 증가한다.

- 설명 (코드 내 작동 순서, 내용 등 추가 설명)

이 함수는 플레이어가 현재 위치에서 어떤 상태인지 체크하는 역할을 한다.

⑦ 목적지에 도착하면 "성공"을 출력하고 종료

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

- 입력 (블록/함수에 입력되는 변수, 값들과 설명)

map: 2차원 배열로, 게임의 맵 상태를 나타낸다

user_x: 현재 플레이어의 x좌표

user_y: 현재 플레이어의 y좌표

- 결과 (블록/함수가 종료된 결과)

finish 변수가 true일 경우: "목적지에 도착했습니다! 축하합니다!"라는 메시지를 출력하고, "게임을 종료합니다."라는 메시지를 출력한 뒤, break 문을 통해 게임 루프를 종료한다..

⑧ 지도와 현재 위치 출력 함수: displayMap()

```
// 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 (블록/함수에 입력되는 변수, 값들과 설명)

int map[][] = 전체 지도

user_x = 유저 x 값

user_y = 유저 y 값

- 반환값 (함수의 경우 작성)

없음

- 결과 (블록/함수가 종료된 결과)

전체 지도를 출력

사용자 위치를 출력

- 설명 (코드 내 작동 순서, 내용 등 추가 설명)

2차원 배열에 있는 맵을 출력

출력하다가 사용자 위치와 동일한 좌표를 발견할 경우 사용자 정보를 출력

⑨ 사용자 위치 체크 함수: checkXY()

```
// 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 (블록/함수에 입력되는 변수, 값들과 설명)

user_x: 사용자(플레이어)의 현재 x좌표

mapX: 맵의 가로 크기 .

user_y: 사용자(플레이어)의 현재 y좌표.

mapY: 맵의 세로 크기 (정수형)

- 반환값 (함수의 경우 작성)

반환값: bool

true: 사용자가 이동하려는 좌표가 유효한 경우.

false: 사용자가 이동하려는 좌표가 유효하지 않은 경우.

- 결과 (블록/함수가 종료된 결과)

함수가 호출된 후, 사용자가 입력한 좌표가 맵의 유효한 범위 내에 있을 경우 true를 반환한다. 그렇지 않으면 false를 반환한다.

- 설명 (코드 내 작동 순서, 내용 등 추가 설명)

checkFlag 변수를 false로 초기화한다.

if 문을 통해 두 가지 조건을 검사:

user_x가 0 이상이고 mapX보다 작은지 확인 ,user_y가 0 이상이고 mapY보다 작은지 확인
두 조건이 모두 참일 경우 checkFlag를 true로 설정한다.

함수는 최종적으로 checkFlag 값을 반환한다.

⑩ 목적지에 도착 체크 함수: checkGoal()

```
// 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 (블록/함수에 입력되는 변수, 값들과 설명)

map: 2차원 정수 배열로, 게임 맵의 상태를 나타낸다

user_x: 사용자(플레이어)의 현재 x좌표

user_y: 사용자(플레이어)의 현재 y좌표.

- 반환값 (함수의 경우 작성)

반환값: bool

true: 사용자가 현재 위치한 좌표가 목적지일 경우

false: 사용자가 현재 위치한 좌표가 목적지가 아닐 경우

- 결과 (블록/함수가 종료된 결과)

함수가 호출된 후, 사용자가 위치한 좌표가 맵의 목적지 좌표인지 확인하고 목적지에 도달했으면 true를 반환한다. 그렇지 않으면 false를 반환한다.

- 설명 (코드 내 작동 순서, 내용 등 추가 설명)

if 문을 사용하여 map 배열의 user_y와 user_x에 해당하는 좌표의 값이 4인지 확인한다. 값이 4일 경우, 해당 좌표는 목적지를 나타낸다. 조건이 참일 경우 함수는 true를 반환하여 목적지에 도달했음을 나타내고 조건이 거짓일 경우 함수는 false를 반환하여 목적지에 도달하지 않았음을 나타낸다.

4. 테스트

- 기능 별 테스트 결과

① 지도 밖으로 나가게 되면 에러 메시지 출력

```
현재 체력: 20 명령어를 입력하세요 (up,down,left,right,map,exit): up
맵을 벗어났습니다. 다시 돌아갑니다.
현재 체력: 20 명령어를 입력하세요 (up,down,left,right,map,exit):
```

② “지도”를 입력하면 전체 지도와 함께 현재 위치를 출력

```
현재 체력: 20 명령어를 입력하세요 (up,down,left,right,map,exit): map
USER |아이템| 적 | |목적지|
-----
아이템| | | |적 | |
-----
| | | | | |
-----
| 적 | 포션 | | |
-----
포션 | | | |적 |
-----
현재 체력: 20 명령어를 입력하세요 (up,down,left,right,map,exit):
```

③ 적을 만났을 때 출력

```
현재 체력: 17 명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다.
|아이템| 적 | |목적지|
-----
아이템| | | |적 | |
-----
| | | | | |
-----
| USER | 포션 | | |
-----
포션 | | | |적 |
-----
적이 있습니다 HP가 2 줄어듭니다.
현재 체력: 14 명령어를 입력하세요 (up,down,left,right,map,exit):
```

④ 포션을 만났을 때 출력

```
현재 체력: 14 명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다.
|아이템| 적 | |목적지|
-----
아이템| | | |적 | |
-----
| | | | | |
-----
| 적 | USER | | |
-----
포션 | | | |적 |
-----
포션이 있습니다 HP가 2 증가합니다.
```


⑤ HP가 0이 되었을 때 출력

```
현재 체력: 1 명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다.
  |아이템| 적 |   |목적지|
-----
아이템|   |   | 적 |   |
-----
      |   | USER |   |   |
-----
      | 적 | 포션 |   |   |
-----
포션 |   |   |   | 적 |
-----
HP가 0이 되었습니다. 실패입니다!
```

- 최종 테스트 스크린샷

```
현재 체력: 14 명령어를 입력하세요 (up,down,left,right,map,exit): up
위로 한 칸 올라갑니다.
  |아이템| 적 |   |목적지|
-----
아이템|   |   | 적 | USER |
-----
      |   |   |   |   |
-----
      | 적 | 포션 |   |   |
-----
포션 |   |   |   | 적 |
-----
현재 체력: 13 명령어를 입력하세요 (up,down,left,right,map,exit): up
위로 한 칸 올라갑니다.
  |아이템| 적 |   | USER |
-----
아이템|   |   | 적 |   |
-----
      |   |   |   |   |
-----
      | 적 | 포션 |   |   |
-----
포션 |   |   |   | 적 |
-----
목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.
```

5. 결과 및 결론

- 프로젝트 결과
 - : mud game에 추가기능 요구사항을 추가해 만들었다.
- 느낀점
 - : c++ 공부를 좀 더 해야겠다고 느꼈다.