

Mud game

224632 조혜민

1. 서론

1. 프로젝트 목적 및 배경: CPP 8주차 프로젝트형 실습 진행 중인 도중 최종 프로젝트형에 도움을 주기 위해 지금까지 배운 걸 활용하여 mud game이란걸 만들어 보고자함.

2. 목표: 창을 통해 게임을 하는 것으로 추가 기능을 넣어 완성하고자 하는게 목표

2. 요구사항

1. 사용자 요구사항: 유저가 상하좌우로만 이동하며 목적지에 도착하는 게임(5x5)

2. 기능 계획

-사용자에게 "상", "하", "좌", "우", "지도", "종료" 중 하나를 입력 받기(상/하/좌/우로 움직인 후 지도 출력하기; 지도를 입력받으면 지도 출력하기; 종료를 누르면 게임이 종료되기;)

-지도 밖으로 나가게 되면 에러메시지 출력

-목적지에 도착하면 "성공"을 출력하고 종료

-유저는 체력을 20을 가지고 시작(이동할때마다 1씩 감소; 처음 명령문과 같이 hp 출력; 0이하가 되면 "실패"를 출력하고 종료;)

3. 함수 계획

-메인함수 : 사용자에게 값을 계속 입력받고, 그에 대한 함수 호출

-displayMap() : 지도와 현재 위치 출력 함수

-checkXY() : 사용자 위치 체크 함수

-checkGoal() : 목적지에 도착 체크 함수

-checkState() : 아이템,포션,적을 만났을 때 출력 함수

3. 설계 및 구현

1. 기능 별 구현 사항: (요구사항 별 코드)

```
-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;
}
}

```

->void displayMap에 대한 함수

함수에 입력되는 변수와 값들은 int map[][]=전체지도, user_x=유저 x 값, user_y=유저 y값
반환값은 void 함수로 존재하지 않는다.

결과는 전체 지도를 출력하고 사용자 위치를 출력하는 결과가 나온다.

if문을 살펴보면 만약 우리가 user가 있을 곳을 이동해서 움직여 주는것이다.

else문을 살펴보면 빈칸,아이템,적,포션,목적지에 어떤 값이 주어져있는지 주는것이다.

따라서 2차원 배열로 만들게 되며 사용자 위치와 동일한 좌표에 있을 때 user 출력하고 빈
칸,아이템,적,포션,목적지에 관한 값에 따른 지도 모양이 출력된다.

```

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;
}

```

```
}
```

-> int user_x=유저 x값,user_y=유저 y값 int mapX=지도 x값, int mapY=지도 y값으로 함수에 입력되는 변수들이다.

반환값은 bool 함수이므로 0 혹은 1이다.

결과는 true 혹은 false가 나오는데

코드를 보면 if값에서 만약 유저위치가 지도 내에 있다면 true로 하고

else문에서 밖으로 나가게 되면 false가 출력되게한다.

```
-bool checkGoal(int map[][mapX], int user_x, int user_y) {  
    // 목적지 도착하면  
    if (map[user_y][user_x] == 4) {  
        return true;  
    }  
    return false;  
}
```

->입력받는 값은 int map[][], int user_x=유저 x값,int user_y=유저 y값이 입력받는 값이다.
반환값은 마찬가지로 bool이므로 0혹은 1이다.

결과는 true 아님 false가 나오는데

if문에서 만약 목적지에 도착한다면 true고

if문이 아니라면 false 가 나오게 된다.

```
-void checkState(int map[][mapX], int user_x, int user_y) {  
    if (user_x == 0 && user_y == 1) {  
        cout << "아이템이 있습니다." << endl;  
    }  
    else if (user_x == 1 && user_y == 0) {  
        cout << "아이템이 있습니다." << endl;  
    }  
    else if (user_x == 0 && user_y == 4) {  
        cout << "포션이 있습니다. HP가 2 상승합니다." << endl;  
        hp += 2;  
    }  
    else if (user_x == 2 && user_y == 3) {  
        cout << "포션이 있습니다. HP가 2 상승합니다." << endl;  
        hp += 2;  
    }  
    else if (user_x == 1 && user_y == 3) {
```

```

        cout << "적이 있습니다. HP가 2 줄어듭니다." << endl;
        hp -= 2;
    }
    else if (user_x == 4 && user_y == 4) {
        cout << "적이 있습니다. HP가 2 줄어듭니다." << endl;
        hp -= 2;
    }
    else if (user_x == 3 && user_y == 1) {
        cout << "적이 있습니다. HP가 2 줄어듭니다." << endl;
        hp -= 2;
    }
}
}

```

->입력은 int map과 int user_x int user_y가 된다

반환값은 없다.

결과는 아이템과 적 그리고 포션을 만나게 된다면 만나게 되었다고 출력하게되고 포션을 만나게 되면 +2, 적을 만나게 되면 -2가 되게된다.

각 위치에 있는 것을 입력하고 뭐가 있는지 알려준다.

```

-int main() {

    // 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

    // 게임 시작
    while (1) { // 사용자에게 계속 입력받기 위해 무한 루프

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

        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 {
    cout << "위로 한 칸 올라갑니다." << endl;
    displayMap(map, user_x, user_y);
    hp--;
    checkState(map, user_x, user_y);
}
}
else if (user_input == "하") {
    // 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;
        displayMap(map, user_x, user_y);
        hp--;
        checkState(map, user_x, user_y);
    }
}
}
else if (user_input == "좌") {
    // 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;
        displayMap(map, user_x, user_y);
        hp--;
        checkState(map, user_x, user_y);
    }
}
}
else if (user_input == "우") {
    // 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;

```

```

        displayMap(map, user_x, user_y);
        hp--;
        checkState(map, user_x, user_y);

    }

}

else if (user_input == "지도") {
    // TODO: 지도 보여주기 함수 호출
    displayMap(map, user_x, user_y);
}

else if (user_input == "종료") {
    cout << "종료합니다.";
    break;
}

else {
    cout << "잘못된 입력입니다." << endl;
    continue;
}

if (hp <= 0) {
    cout << "HP가 0 이하가 되었습니다. 실패했습니다." << endl;
    cout << "게임을 종료합니다." << endl;
    break;
}

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

}

return 0;
}

```

->메인함수로 map을 어디에 뭐가 있는지 입력해준다 그리고 while문을 통해 계속 입력을 받는다 user_input 을 통해 사용자의 입력을 받게하는 변수를 만든다. 그리고 상을 입력하면 위로 한칸 올라가고 checkXY함수를 통해 지도 내에 있는지 체크한다 맵을 벗어났으면 +1을 하여 다시 돌아오게 되고 만약 지도 내에 있다면 상으로 이동하면서 지도를 다시 재정비 하고 checkState를 통해 아이템등이 있는지 체크한다. 좌 우 하 도 마찬가지로 진행한다.

그리고 지도를 입력하게 되면 지도를 입력하는 함수인 displayMap을 보여준다

만약 종료를 입력하게 된다면 종료하고 break로 멈춘다

만약 다른 값이 들어온다면 잘못된 입력이라고 다시 입력하게 한다.

만약 그리고 hp가 0이하라면 게임을 종료하게 된다.

그리고 checkGoal을 통해 목적지에 도착했는지 체크하게 된다.

4. 테스트

- (테스트 : 입력에 따른 원하는 결과나 나오는지 확인하는 과정)

1. 기능 별 테스트 결과: (요구사항 별 스크린샷)

-사용자에게 "상", "하", "좌", "우", "지도", "종료" 중 하나를 입력 받기(상/하/좌/우로 움직인 후 지도 출력하기; 지도를 입력받으면 지도 출력하기; 종료를 누르면 게임이 종료되기;)

하 일 때

```
현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): 하
위로 한 칸 내려갑니다.
  |아이템|  적  |      |목적지|
-----
USER |      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
아이템이 있습니다.
```

상일 때

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

우일 때

현재 HP: 18 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우

오른쪽으로 이동합니다.

USER	적	목적지
아이템	적	
	적	포션
포션		적

좌일때

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

왼쪽으로 이동합니다.

USER	아이템	적	목적지
아이템		적	
	적	포션	
포션			적

지도

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

가이드를 불러옵니다.

USER	아이템	적	목적지
아이템		적	
	적	포션	
포션			적

현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료):

종료를 입력하세요.

종료

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

종료를 입력합니다.

현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): 상
맵을 벗어났습니다. 다시 돌아옵니다.

```

현재 HP: 17 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 이동합니다.
      |아이템|  적  |      | USER |
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.
  
```

```

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

```

성공했을 때

현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 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 이동합니다.

	아이템	USER		목적지
아이템			적	
	적	포션		
포션				적

현재 HP: 18 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 이동합니다.

	아이템	적	USER	목적지
아이템			적	
	적	포션		
포션				적

현재 HP: 17 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 이동합니다.

	아이템	적		USER
아이템			적	
	적	포션		
포션				적

목적지에 도착했습니다! 축하합니다!

게임을 종료합니다.

실패 했을 때

```
Microsoft Visual Studio 디버그 콘솔
현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): 좌
맵을 벗어났습니다. 다시 돌아갑니다.
현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 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 이동합니다.
  |아이템| USER |   목적지 |
-----
아이템|   |   |   적   |   |
-----
  |   |   |   |   |
-----
  |   적   | 포션 |   |   |
-----
포션 |   |   |   |   적   |
-----
현재 HP: 18 명령어를 입력하세요 (상,하,좌,우,지도,종료): 하
위로 한 칸 내려갑니다.
  |아이템|   적   |   목적지 |
-----
아이템|   | USER |   적   |   |
-----
  |   |   |   |   |
-----
  |   적   | 포션 |   |   |
-----
포션 |   |   |   |   적   |
-----
```

생략

```

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

```

5. 결과 및 결론

1. 프로젝트 결과: mud game을 만들어 상하좌우 (5x5)배열에 대해 만들었고 아이템을 만나면 아이템을 먹었다 하고 포션을 먹으며 2가 증가되고 적을 만나면 2가 감소되어 목적지까지 가는 것이다.

2. 느낀 점: 전역변수와 지역변수에 대해 내가 무지하고 있다는 사실을 깨달았다. 좀 더 공부해야겠다고 생각이 들었고 초반에 많이 알려주셔서 할수 있었지만 처음부터 만들라하면 힘들거 같았다. 좀 더 공부하고 싶은 생각이 들었다.