

목차

1.서론

- 프로젝트 목적 및 배경
- 목표 : 몇 가지 조건이 있는 Mud 게임 구현

2.요구사항

- 사용자 요구사항
- 기능 요구사항
- 함수 요구사항

3.설계 및 구현

- 기능 별 구현 사항 (요구사항 별 코드)

4.테스트

- 기능 별 테스트 결과 (요구사항 별 스크린 샷)
- 최종 테스트 스크린 샷 (프로그램 전체 동작 스크린 샷)

5.결과 및 결론

- 프로젝트 결과
- 느낀 점

1.서론

-프로젝트 목적 및 배경

이번 시간 이전까지 수업 시간에 배웠던 c++의 프로그램 기본 구조, 자료 형과 변수(선언, 초기화), 상수, 출력함수 cout, 입력함수 cin, 연산자와 표현식, 조건문(if-else, switch 문), 반복문 (for 문, while 문, do-while 문), 배열, 2차원 배열 등을 배웠고 이 내용에 대한 실습을 위해 진행하게 되었다.

-목표

Mud 게임 구현

2.요구 사항

-사용자 요구사항

: 유저가 상하좌우로만 이동하며 목적지에 도착하는 게임

-기능 요구사항

1)사용자에게 “상”, “하”, “좌”, “우”, “지도”, “종료” 중 하나를 입력 받기

-상/하/좌/우 입력시 해당 방향으로 이동 후 지도 출력

-“지도”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)목적지에 도착하면 “성공”을 출력하고 종료

-함수 계획

1) 메인 함수 : 사용자에게 값을 계속 입력 받고, 그에 대한

함수 호출

- 2)지도와 현재 위치 출력 함수 : displayMap()
- 3)사용자 위치 체크 함수 : checkXY()
- 4)목적지에 도착 체크 함수 : checkGoal()
- 5)위치에 따른 HP 변화 출력 함수 : checkState()

-추가 기능 요구사항

- 1)유저는 체력 20을 가지고 게임 시작
- 2)사용자가 이동할 때 마다 사용자 체력 1씩 감소
- 3)처음 명령문을 입력 받을 때 마다 HP 함께 출력
- 3)HP가 0이 되면 “실패”를 출력하고 종료
- 5)무기/갑옷, 포션, 적을 만났을 때 그에 대한 메시지를 출력

3. 설계 및 구현

*기능 별 구현 사항

- 1)사용자에게 “상”, “하”, “좌”, “우”, “지도”, “종료” 중 하나를 입력 받기

```
// 사용자의 입력을 저장할 변수 선언
string user_input = "";
cout << "명령어를 입력하세요 (상,하,좌,우,지도,종료): ";
cin >> user_input;
```

-입력

없음

-결과

입력 받은 값을 user_input에 넣어준다.

-설명

사용자에게 여러 선택지 중에 입력을 받고 그 값을 변수에 저장

2)지도 밖으로 나가게 되면 에러 메시지 출력

```
if (inMap == false) {  
    cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;  
    user_x += 1;  
}
```

-입력

“상하좌우” 중에서 입력을 받는다.

-결과

조건에 맞지 않을 시 특정 문장을 출력

-설명

입력을 받았으나 만약 그게 지도 밖으로 나가는 입력을 했을 경우에 해당해서 문장을 출력한다.

3)목적지에 도착하면 “성공”을 출력하고 종료

```
// 목적지에 도달했는지 체크  
//finish를 논리형으로 선언, checkGoal 가 true 이면 아랫 문장 출력  
bool finish = checkGoal(map, user_x, user_y);  
if (finish == true) {  
    cout << "목적지에 도착했습니다! 축하합니다!" << endl;  
    cout << "게임을 종료합니다." << endl;  
    break;  
}
```

-입력

“상하좌우” 중에서 입력을 받는다.

-결과

‘성공’을 출력하고 게임을 종료

-설명

사용자가 입력한 후에 목적지에 도착하여 도착했다는 문장을 출력하고 또한 게임을 종료한다는 문장을 출력한다.

*함수 별 구현 사항

1) 메인 함수

```
// 메인 함수
int main() {
    // 0은 빈 공간, 1은 아이템, 2는 적, 3은 포션, 4는 목적지 으로 배열 선언
    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 선언 및 초기화
    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 선언
        string user_input = "";
        cout << "명령어를 입력하세요 (상,하,좌,우,지도,종료): ";
        cin >> user_input;

        if (user_input == "상") {
            // 위로 한 칸 올라가기
            user_y -= 1;
            //위로 한칸 올라간 것이 벗어났는지 아닌지 체크
            bool inMap = checkXY(user_x, mapX, user_y, mapY);
            if (inMap == false) {
                cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
                user_y += 1;
            }
            //벗어나지 않았다면 출력
        } else {
            cout << "위로 한 칸 올라갑니다." << endl;
            displayMap(map, user_x, user_y);
        }
    }
}
```

-입력

상하좌우지도종료 중에서 하나를 입력

-반환값

상하좌우지도종료에 해당하는 값이 출력되거나 만약 지도에서 벗어나는 경우에는 벗어났다는 값 출력

-결과

사용자가 한칸씩 움직이거나 다시 입력하라하거나 종료 되거나
등등 해당하는 문장이 출력

-설명

입력값에 따라 나오는 결과에 맞는 함수 값을 반환해준다.

2)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 1"; // 양 옆 1칸 공백
            }
            else {
                int posState = map[i][j];
                switch (posState) {
                    case 0:
                        cout << "      1"; // 6칸 공백
                        break;
                    case 1:
                        cout << "아이템 1";
                        break;
                    case 2:
                        cout << " 적   1"; // 양 옆 2칸 공백
                        break;
                    case 3:
                        cout << " 포션 1"; // 양 옆 1칸 공백
                        break;
                    case 4:
                        cout << "목적지 1";
                        break;
                }
            }
        }
        cout << endl;
        cout << " ----- " << endl;
    }
}
```

-입력

map : 전체 지도

user_x : 유저 x값

user_y : 유저 y값

-반환값

없음

-결과

전체 지도를 출력하고 사용자 위치를 출력

-설명

2차원 배열의 지도 출력, 출력할 때 사용자 위치와 동일한 좌표일 경우에 사용자 정보를 출력

3)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;
    //아래의 조건이 맞다면 true 값
    if (user_x >= 0 && user_x < mapX && user_y >= 0 && user_y < mapY) {
        checkFlag = true;
    }
    return checkFlag;
}
```

-입력

user_x : 유저 x값

user_y : 유저 y값

mapx : 지도 x값

mapy : 지도 y 값

-반환값

true or false

-결과

true나 false가 반환되어 해당하는 과정으로 진행

-설명

지도를 벗어났는지 체크하기 위해 사용자의 위치를 체크하는 함수

4)목적지에 도착 체크 함수 : checkGoal()

```
// 유저의 위치가 목적지인지 체크하는 함수
bool checkGoal(int map[][mapX], int user_x, int user_y) {
    // 목적지 도착하면 true 값 반환
    if (map[user_y][user_x] == 4) {
        return true;
    }
    return false;
}
```

-입력

map : 전체 지도

user_x : 유저 x값

user_y : 유저 y값

-반환값

true or false

-결과

값이 목적지에 해당하는 경우 true 출력

-설명

유저의 위치가 목적지인지 확인하기 위한 함수

5)위치에 따른 HP 변화 출력 함수 : checkState()

-입력

map : 전체 지도

user_x : 유저 x값

user_y : 유저 y값

-반환값

해당하는 경우의 case값


```

// 0은 빈 공간, 1은 아이템, 2는 적, 3은 포션, 4는 목적지
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;
    }
}

```

-결과

map의 입력 값에 따라 해당하는 경우의 문장 출력

-설명

0은 빈 공간, 1은 아이템, 2는 적, 3은 포션, 4는 목적지

0번 그냥 체력 1이 떨어진다.

1번 아이템이 있습니다

2,3번 은 hp의 변화까지 출력

4번은 목적지에 도착했다고 출력

4. 테스트

*기능 별 테스트 결과

1)사용자에게 “상”, “하”, “좌”, “우”, “지도”, “종료” 중 하나를 입력 받기

현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료):

2)지도 밖으로 나가게 되면 에러 메시지 출력

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

3)목적지에 도착하면 “성공”을 출력하고 종료

목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.

*함수 별 테스트 결과

1) 메인 함수 : 사용자에게 값을 계속 입력 받고, 그에 대한
함수 호출

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

2)지도와 현재 위치 출력 함수 : displayMap()

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

3)사용자 위치 체크 함수 : checkXY()

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

4)목적지에 도착 체크 함수 : checkGoal()

```

|아이템|적|USER|
-----
아이템|   |   |적|   |
-----
|   |   |   |   |
-----
|   |적|포션|   |
-----
포션|   |   |   |적|
-----
목적지에 도착했습니다! 축하합니다!

```

5)위치에 따른 HP 변화 출력 함수 : checkState()

```

현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 이동합니다.
|아이템|적|목적지|
-----
아이템|   |   |USER|   |
-----
|   |   |   |   |   |
-----
|   |적|포션|   |   |
-----
포션|   |   |   |   |적|
-----
적이 있습니다. HP가 2 줄어듭니다.
현재 HP: 18 명령어를 입력하세요 (상,하,좌,우,지도,종료):

```

*최종 테스트 결과

```

현재 HP: 18 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 이동합니다.
|아이템|적|목적지|
-----
아이템|   |   |적|USER|   |
-----
|   |   |   |   |   |
-----
|   |적|포션|   |   |
-----
포션|   |   |   |   |적|
-----
현재 HP: 18 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
맵을 벗어났습니다. 다시 돌아갑니다.
현재 HP: 18 명령어를 입력하세요 (상,하,좌,우,지도,종료): 상
위로 한 칸 올라갑니다.
|아이템|적|USER|
-----
아이템|   |   |적|   |
-----
|   |   |   |   |   |
-----
|   |적|포션|   |   |
-----
포션|   |   |   |   |적|
-----
목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.

```

5. 결과 및 결론

결과 : mud게임을 만들었다.