

c++프로그래밍 및 실습

Mud game

프로젝트 보고서

제출일자: 2024.11.03

제출자명: 윤동해

제출자학번: 234129

목차

1. 서론

- 1) 프로젝트 목적 및 배경
- 2) 목표

2. 요구사항

- 1) 사용자 요구사항
- 2) 기능 계획
- 3) 함수 계획

3. 설계 및 구현

- 1) 기능 별 구현 사항

4. 테스트

- 1) 기능 별 테스트 결과
- 2) 최종 테스트 스크린샷

5. 결과 및 결론

- 1) 프로젝트 결과: mudgame을 만들었다.
- 2) 느낀 점

1. 서론

1) 프로젝트 목적 및 배경

Data Type, Variable, In out, operator, condition, Loop, Array 및 function을 학습한 내용을 바탕으로 간단한 Mudgame을 구현한다.

2) 목표

상하좌우로 이동하며 무사히 목적지에 도착하게 게임을 구현한다.

2. 요구사항

1) 사용자 요구사항

- ① 유저는 체력 20을 가지고 게임 시작
- ② 사용자가 이동할 때마다 사용자 체력 1씩 감소
- ③ 처음 명령문을 입력받을 때마다 HP 함께 출력
- ④ HP가 0이 되면 "실패"를 출력하고 종료
- ⑤ 무기/갑옷, 포션, 적을 만났을 때 그에 대한 메시지를 출력

2) 기능 계획

- ① 사용자에게 "상", "하", "좌", "우", "지도", "종료" 중 하나를 입력 받기
 - 상/하/좌/우 입력 시 해당 방향으로 이동 후 지도 출력
 - "지도"를 입력하면 전체 지도와 함께 현재 위치를 출력 (체력 감소 없음)
 - 이 중 다른 것을 입력하면 에러 메시지 출력 후 재입력 요청
- ② 지도 밖으로 나가게 되면 에러 메시지 출력
- ③ 목적지에 도착하면 "성공"을 출력하고 종료

3) 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 현재 체력 상태 체크 함수: currentStatus()

3. 설계 및 구현

1) 기능 별 구현 사항

```
// 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 |"; // 알 옆 1칸 공백
            } else {
                int posState = map[i][j];
                switch (posState) {
                    case 0:
                        cout << "   |"; // 6칸 공백
                        break;
                    case 1:
                        cout << " 걸 |"; // 알 옆 2칸 공백
                        break;
                    case 2:
                        cout << " 적 |"; // 알 옆 2칸 공백
                        break;
                    case 3:
                        cout << " 포션 |"; // 알 옆 1칸 공백
                        break;
                    case 4:
                        cout << "목표지|";
                        break;
                    case 5:
                        cout << " 갑옷 |"; // 알 옆 1칸 공백
                        break;
                }
            }
        }
        cout << endl;
        cout << "-----" << endl;
    }
}
```

```
// 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;
}
```

1. 함수 스크린샷 (좌)
2. 입력
 - int map[][] = 전체 지도
 - user_x = 유저 x 값
 - user_y = 유저 y 값
3. 반환값
 - 없음
4. 결과
 - 전체 지도를 출력
 - 사용자 위치를 출력
5. 설명
 - 2차원 배열에 있는 맵을 출력
 - 출력하다가 사용자 위치와 동일한 좌표를 발견할 경우 사용자 정보를 출력

1. 함수 스크린샷 (좌)
2. 입력
 - int user_x: 사용자가 이동하려는 x 좌표 (가로)
 - int mapX: 맵의 가로 크기
 - int user_y: 사용자가 이동하려는 y 좌표 (세로)
 - int mapY: 맵의 세로 크기
3. 반환값
 - true: 사용자가 이동하려는 좌표가 맵의 유효한 범위 내에 있을 때
 - false: 사용자가 이동하려는 좌표가 맵의 유효한 범위를 벗어날 때
4. 결과
 - true 또는 false를 반환
5. 설명
 - user_x와 user_y가 각각 0 이상이고 맵의 크기 mapX와 mapY보다 작은

지 확인

- 조건문이 참이면 checkFlag 변수에 true 값을 할당
- 조건문이 거짓이면 checkFlag 변수는 false 값을 유지
- checkFlag 변수를 반환

```
// 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;
}
```

1. 함수 스크린샷 (좌)
2. 입력
 - int map[][mapX]: 맵의 특정 위치를 나타냄
 - int user_x: 유저의 현재 x 좌표
 - int user_y: 유저의 현재 y 좌표
3. 반환값
 - true: 유저의 현재 위치가 목적지일 때
 - false: 유저의 현재 위치가 목적지가 아닐 때
4. 결과
 - true 또는 false를 반환
5. 설명
 - 유저의 현재 위치 값 확인
 - 조건문이 참이면 true 값을 반환
 - 조건문이 거짓이면 false 값을 반환

```
// 현재 상태를 출력하는 함수
void currentStatus(int hp) {
    cout << "현재 HP: " << hp << endl;
}
```

1. 함수 스크린샷 (좌)
2. 입력
 - int hp: 유저의 현재 체력 (HP).
3. 반환값
 - 없음
4. 결과
 - 함수는 현재 체력(HP)을 출력
5. 설명
 - 유저의 현재 체력을 hp 변수를 통해 출력

현재 HP: 20
명령어를 입력하세요 (상,하,좌,우,지도,종료): wleh
작무된 인력입니다.

2) 최종 테스트 스크린샷

현재 HP: 20
명령어를 입력하세요 (상,하,좌,우,지도,종료): dn
잘못된 입력입니다.

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

| | USER | 적 | | 목적지 |
|----|------|----|---|-----|
| 갑옷 | | | 적 | |
| | | | | |
| | 적 | 포션 | | |
| 포션 | | | | 적 |

검이 있습니다!

현재 HP: 19
명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 이동합니다.

| | 검 | USER | | 목적지 |
|----|---|------|---|-----|
| 갑옷 | | | 적 | |
| | | | | |
| | 적 | 포션 | | |
| 포션 | | | | 적 |

적이 있습니다! 전투가 발생하여 HP가 2 감소합니다.

현재 HP: 16
명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 이동합니다.

| | 검 | 적 | USER | 목적지 |
|----|---|----|------|-----|
| 갑옷 | | | 적 | |
| | | | | |
| | 적 | 포션 | | |
| 포션 | | | | 적 |

현재 HP: 15
명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 이동합니다.

| | 검 | 적 | | USER |
|----|---|----|---|------|
| 갑옷 | | | 적 | |
| | | | | |
| | 적 | 포션 | | |
| 포션 | | | | 적 |

목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.

5. 결과 및 결론

1) 프로젝트 결과

mud game을 요구사항에 맞춰 만들었다.

2) 느낀 점

기말고사 대체 프로젝트처럼 무에서 유를 창출할 때는 어려운데 이번 mud game 프로젝트처럼 기본 코드가 제공될 때는 이제는 할만해진 것 같다. 사실 중간고사까지는 어려움을 겪었으나 중간 리뷰와 혼자 공부를 통해서 많이 배워 전보다는 나은 실력으로 이번 활동을 마무리할 수 있어서 즐겁다. 이제 기말고사 프로젝트를 시작할 생각을 하니 조금 두려워졌다.