보고서 작성 (1/4)

1. 서론

1. 프로젝트 목적 및 배경: 7주차까지 배운 내용에 대한 복습겸 실습

2. 목표: 요구사항에 대한 간단한 MUD게임 구현

2. 요구사항

1. 사용자 요구사항: 유저가 상하좌우로만 이동하며 목적지에 도착하는 게임

2. 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

2-1. 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

④ HP가 0이 되면 "실패"를 출력하고 종료

⑤ 무기/갑옷, 포션, 적을 만났을 때 그에 대한 메시지를 출력

• 예) {X}가 있습니다.

• 적을 만날 경우 HP가 2가 줄어들고 그에 대한 추가 메시지 출력

3. 설계 및 구현

1. 기능 별 구현 사항: (요구사항 별 코드)

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

```
// 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;
    }
}
```

1. 함수 스크린샷 (좌)

2. 입력 (블록/함수에 입력되는 변수, 값들과 설명)

- int map[][] = 전체 지도
- user_x = 유저 x 값
- user_y = 유저 y 값

3. 반환값 (함수의 경우 작성)

- 없음

4. 결과 (블록/함수가 종료된 결과)

- 전체 지도를 출력
- 사용자 위치를 출력

5. 설명 (코드 내 작동 순서, 내용 등 추가 설명)

6. 2차원 배열에 있는 맵을 출력

7. 출력하다가 사용자 위치와 동일한 좌표

를 발견할 경우 사용자 정보를 출력

```
// 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 (상단)

2. 입력

int user_x: 사용자가 이동하려는 x 좌표 (가로 위치).

int mapX: 맵의 가로 크기 (최대 x 좌표).

int user_y: 사용자가 이동하려는 y 좌표 (세로 위치).

int mapY: 맵의 세로 크기 (최대 y 좌표).

3. 반환값

없음:

4. 결과::사용자가 이동하고자 하는 좌표가 유효한 경우 true를 반환하고, 그렇지 않은 경우 false를 반환한다

5. 설명

함수는 checkFlag라는 불리언 변수를 false로 초기화하여 유효성 체크 결과를 저장한다

사용자 x 좌표(user_x)가 0 이상이고, 맵의 가로 크기(mapX)보다 작은지 검사하고 사용자 y 좌표(user_y)가 0 이상이고, 맵의 세로 크기(mapY)보다 작은지 검사한다

두 조건이 모두 만족되면, checkFlag를 true로 설정하여 유효한 좌표임을 나타낸다

마지막으로 checkFlag 값을 반환하여 호출자에게 결과를 전달한다

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

1. 함수 스크린샷 (상단)

2. 입력

int map[][mapX]: 게임 맵을 나타내는 2차원 배열.

int user_x: 사용자의 현재 x 좌표 (가로 위치).

int user_y: 사용자의 현재 y 좌표 (세로 위치).

3. 반환값

4. 결과

사용자의 현재 위치가 목적지(값이 4인 위치)인지 확인하여, 해당 결과에 따라 true 또는 false를 반환한다

5. 설명

코드 내 작동 순서:

그 위치의 값이 4인지 비교하여, 만약 값이 4라면 true를 반환하여 사용자가 목적지에 도달했음을 알려주며

값이 4가 아니면 false를 반환하여 목적지에 도달하지 않았음을 알려준다

```cpp
// 유저의 위치에 있는 아이템에 따른 상호작용 및 HP 변경 함수
void checkState(int map[][mapX], int user_x, int user_y) {
    int posState = map[user_y][user_x];
    if (posState == 1) {
        cout << "아이템이 있습니다." << endl;
    } else if (posState == 2) {
        user_hp -= 2; // 적을 만나면 체력 2 감소
        cout << "적을 만났습니다! HP가 2 감소합니다." << endl;
    } else if (posState == 3) {
        user_hp += 2; // 포션을 만나면 체력 2 증가
        cout << "포션을 발견했습니다! HP가 2 증가합니다." << endl;
    }
}
```

1. 함수스크린샵(상단)

2. 입력

입력되는 변수 및 값들:

int map[][mapX]: 게임 맵을 나타내는 2차원 배열.

int user_x: 사용자의 현재 x 좌표 (가로 위치).

int user_y: 사용자의 현재 y 좌표 (세로 위치).

3. 반환값

해당 위치에 따른 메시지를 나타내며 user_hp의 값을 증감시킨다

4. 결과

아이템이 있는 경우: "아이템이 있습니다." 메시지 출력.

적이 있는 경우: HP가 2 감소하고, "적을 만났습니다! HP가 2 감소합니다." 메시지 출력.

포션이 있는 경우: HP가 2 증가하고, "포션을 발견했습니다! HP가 2 증가합니다." 메시지 출력. 를 나타낸다

5. 설명

함수는 사용자 위치의 맵 상태를 posState 변수에 저장하며 그 값에 따라 해당하는 조건문을 수행한다

아이템(값이 1): 아이템이 있는 경우 "아이템이 있습니다." 를 출력

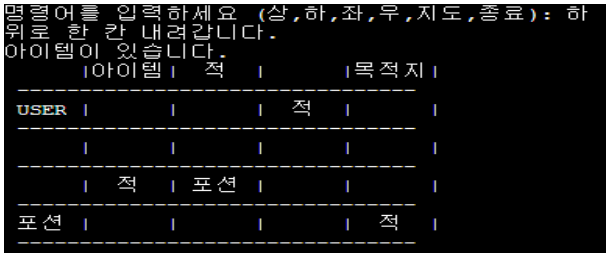
적(값이 2): 적을 만났을 경우, HP를 2 감소시키고 "적을 만났습니다! HP가 2 감소합니다." 라는 메시지를 출력

포션(값이 3): 포션을 발견했을 경우, HP를 2 증가시키고 "포션을 발견했습니다! HP가 2 증가합니다."라는 메시지를 출력.

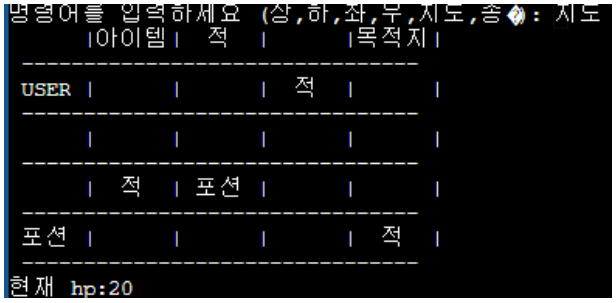
4. 테스트

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

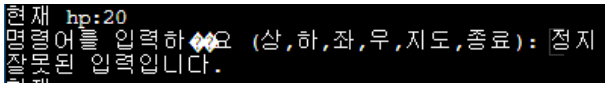
- 상/하/좌/우 입력시 해당 방향으로 이동 후 지도 출력



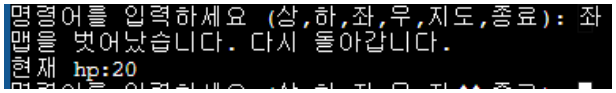
•"지도"를 입력하면 전체 지도와 함께 현재 위치를 출력



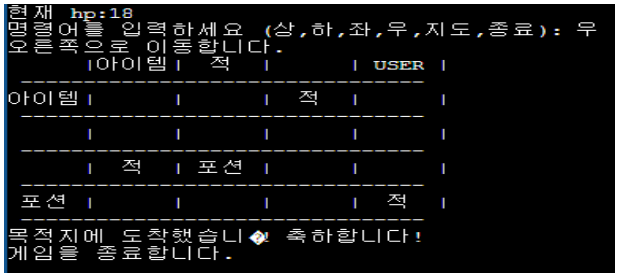
• 이 중 다른 것을 입력하면 에러 메시지 출력 후 재 입력 요청



② 지도 밖으로 나가게 되면 에러 메시지 출력

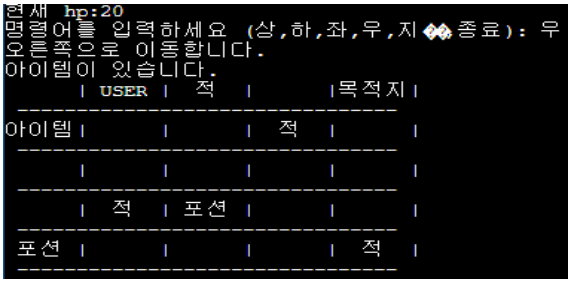


③ 목적지에 도착하면 "성공"을 출력하고 종료

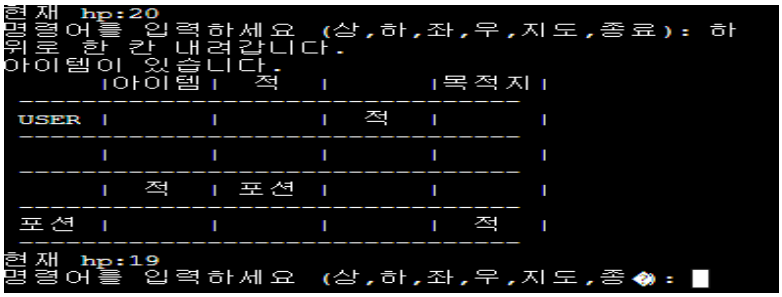


2-1. 추가 기능 요구사항

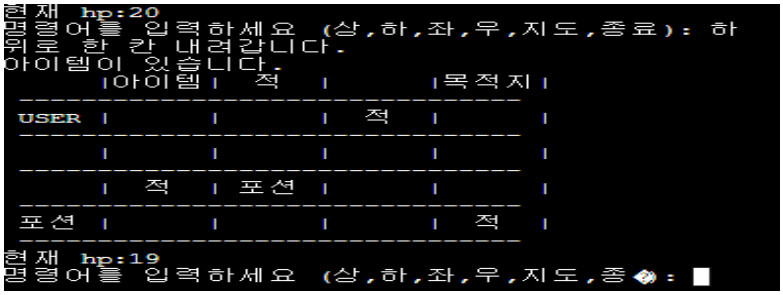
① 유저는 체력 20을 가지고 게임 시작



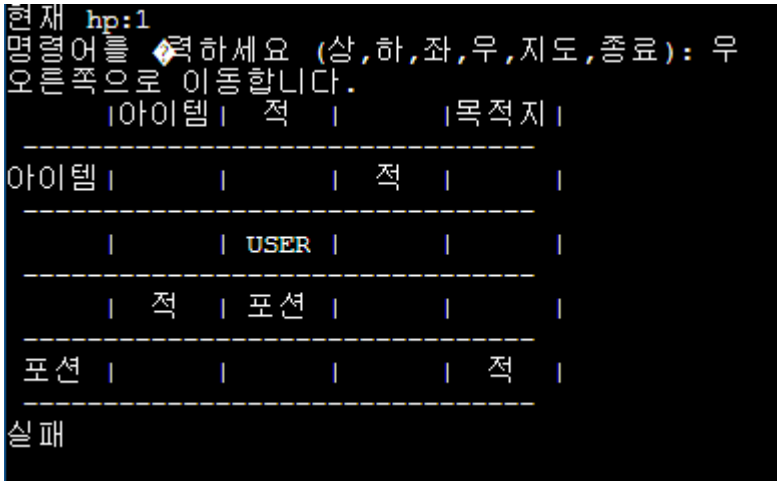
② 사용자가 이동할 때 마다 사용자 체력 1씩 감소



③ 처음 명령문을 입력 받을 때 마다 HP 함께 출력



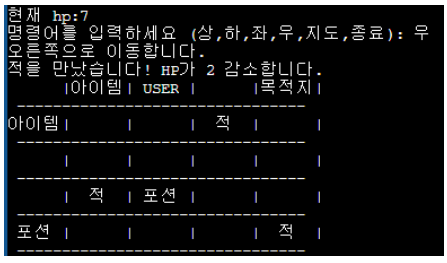
④ HP가 0이 되면 "실패"를 출력하고 종료

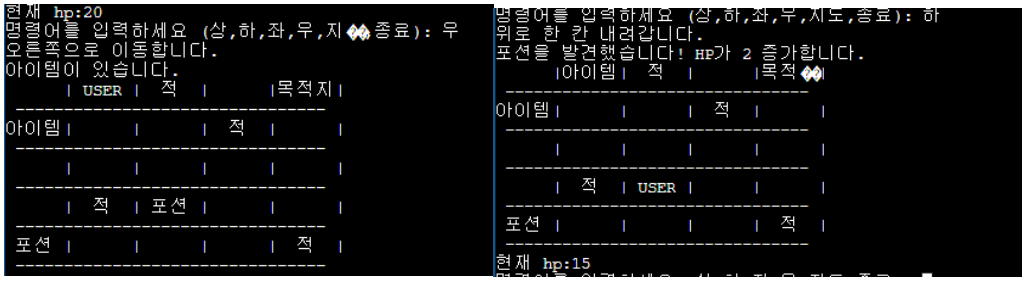


⑤ 무기/갑옷, 포션, 적을 만났을 때 그에 대한 메시지를 출력

- 예) {X}가 있습니다.

- 적을 만날 경우 HP가 2가 줄어들고 그에 대한 추가 메시지 출력





4-2. 최종 테스트 스크린샷: (프로그램 전체 동작 스크린샷)

```
현재 hp:20
명령어를 입력하세요 (상,하,좌,우,지 종료): 우
오른쪽으로 이동합니다.
아이템이 있습니다.
| USER | 적 | 목적지 |
 ---------------------------
아이템 | 적 |
 ---------------------------
 ---------------------------
 적 | 포션 |
 ---------------------------
 포션 | 적 |
 ---------------------------
현재 hp:19
명령어를 입력하세요 (상,하,좌,우,지도,종 : 우
오른쪽으로 이동합니다.
적을 만났습니다! HP가 2 감소합니다.
|아이템| USER | 목적지 |
 ---------------------------
아이템 | 적 |
 ---------------------------
 ---------------------------
 적 | 포션 |
 ---------------------------
 포션 | 적 |
 ---------------------------
현재 hp:16
명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 이동합니다.
|아이템| 적 | USER | 목적지 |
 ---------------------------
아이템 | 적 |
 ---------------------------
 ---------------------------
 적 | 포션 |
 ---------------------------
 포션 | 적 |
 ---------------------------
현재 hp:15
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.
```

5. 결과 및 결론

1. 프로젝트 결과: 몇 가지 추가 기능사항을 적용시킨mud game을 만들었다.

2. 느낀 점: 이번 mud게임을 만들면서 user_hp를 초반에 지역변수로 잘못 설정하여 어떠한 행동을 취해도 user_hp가 20에서 바뀌지 않는 상황이 답답했는데 차분하게 코드를 들여다보니 내가 잘못 설정한 것을 발견했다. 천천히 코드 하나 살펴보는 것도 중요한 것 같다. 또한 다양함 함수를 설정하고 호출하는 과정에서 오랜시간 고민하고 여러가지 예시들을 찾아봤던 과정이 나에게 도움을 주고 그동안 배운 내용을 더욱 몸에 익숙하게 되는 과정이었다고 생각된다

다만 내가 작성한 코드로 visual studio code로 작성시켰을 때 상하좌우가 어떤 단어를 입력해도 잘못된 입력이라고 뜨는데 그 이유를 찾지 못해 온라인 개발 환경 https://www.onlinegdb.com/ 이사이트를 이용하여 결과를 테스트해보았다.