

1. 서론

1. 프로젝트 목적 및 배경: 7주차까지 배운 내용에 대한 실습을 위해 진행
2. 목표: 간단한 Mud 게임 구현 (기본 베이스 코드에 체력 기능을 도입하고, 아이템-적과 상호작용하는 기능을 추가함)

2. 요구사항

1. 사용자 요구사항: player가 상, 하, 좌, 우, 지도, 종료 중 하나를 입력하여 상호작용하는 게임

2. 기능 계획 :

- 지도를 출력하고 벗어나는 경우 경고를 하는 기능
- 상, 하, 좌, 우 입력에 따라 위치를 이동하는 기능(이동 할 때마다 체력이 1씩 마이너스), 종료 입력시
- 아이템에 따라 체력에 영향을 주는 기능(+안내 메시지 기능)
- 체력이 0이하가 되면 게임을 종료하는 기능
- 목적지에 도달하면 게임을 종료하는 기능

3. 함수 계획 : displayMap (맵을 print하는 함수), checkXY (player가 맵 밖에 있는지 확인하는 함수), checkGoal (목적지에 도달했는지 확인하는 함수), handlePositionEffect (아이템-적과 상호작용하는 함수), movePlayer (사용자를 이동하는 함수)

3. 설계 및 구현

-지도를 출력하고 벗어나는 경우 경고를 하는 기능

```
// 맵의 크기
const int mapX = 5;
const int mapY = 5;
// 메인 함수
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}
    };
};
```

(초기 지도 설정값)

```
else if (user_input == "지도") displayMap(map, user_x, user_y);
```

(사용자 입력이 지도인 경우 displayMap 함수 호출)

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

(displayMap 함수로 지도 출력)

-상, 하, 좌, 우 입력에 따라 위치를 이동하는 기능(이동 할 때마다 체력이 1씩 마이너스), 종료 입력시

```
// 사용자의 입력을 저장할 변수
string user_input;
// 사용자의 현재 상태와 입력 요청
cout << "현재 HP: " << hp << " 명령어를 입력하세요 (상,하,좌,우,지도,종료): ";
cin >> user_input;

// 위치 입력에 따른 상호작용
if (user_input == "상") movePlayer(map, hp, user_x, user_y, 0, -1);
else if (user_input == "하") movePlayer(map, hp, user_x, user_y, 0, 1);
else if (user_input == "좌") movePlayer(map, hp, user_x, user_y, -1, 0);
else if (user_input == "우") movePlayer(map, hp, user_x, user_y, 1, 0);
// 지도 보여주기 함수 호출
else if (user_input == "지도") displayMap(map, user_x, user_y);

// 종료를 입력한 경우
else if (user_input == "종료") {
    cout << "종료합니다." << endl;
    break;
} else {
    cout << "잘못된 입력입니다." << endl;
}
```

(위치 입력의 경우 movePlayer 함수 호출 / 지도 입력의 경우 displayMap 함수 호출)

```
//player 이동 함수 (기존 코드의 반복되는 부분을 함수화 했습니다)
void movePlayer(int map[][mapX], int &hp, int &x, int &y, int dx, int dy) {
    // 이동한 위치
    int newX = x + dx;
    int newY = y + dy;
    if (checkXY(newX, newY)) {
        x = newX;
        y = newY;
        hp -= 1; // 체력 -1
        displayMap(map, x, y); //맵에 이동한 위치 표시
        handlePositionEffect(map, hp, x, y); // 아이템-적과 만났을 때 상호작용
    } else {
        cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl; //맵을 벗어나는 입력을 받은 경우
    }
}
```

(유효한 위치인지 확인하는 checkXY함수 호출/ 아이템-적과 만났을 때 handlePositionEffect 함수 호출)

```
// 이동하려는 곳이 유효한 좌표인지 체크하는 함수
bool checkXY(int x, int y) {
    return (x >= 0 && x < mapX && y >= 0 && y < mapY);
}
```

-아이템에 따라 체력에 영향을 주는 기능(+안내 메시지 기능)

```
// 아이템, 적, 포션을 만났을 때
void handlePositionEffect(int map[][mapX], int &hp, int x, int y) {
    switch (map[y][x]) {
        case 1: //아이템을 만난 경우
            cout << "아이템이 있습니다." << endl;
            break;
        case 2: //적을 만난 경우
            cout << "적이 있습니다. HP가 2 줄어듭니다." << endl;
            hp -= 2;
            break;
        case 3: //포션을 만난 경우
            cout << "포션이 있습니다. HP가 2 증가합니다." << endl;
            hp += 2;
            break;
        default: //이외의 경우
            break;
    }
}
```

-체력이 0이하가 되면 게임을 종료하는 기능

```
// 게임 시작
while (true) { // 사용자에게 계속 입력받기 위해 무한 루프
    // 게임 개선시키기 : 체력 기능 도입
    if (hp <= 0) {
        cout << "HP가 0 이하가 되었습니다.\n게임을 종료합니다." << endl;
        break;
    }
}
```

(loop의 가장 처음에 체력이 0 이하인지 확인하는 코드가 있어야 됩니다)

-목적지에 도달하면 게임을 종료하는 기능

```
// 목적지에 도달한 경우
if (checkGoal(map, user_x, user_y)) {
    cout << "목적지에 도착했습니다! 축하합니다!\n게임을 종료합니다." << endl;
    break;
}
```

(목적지에 도달한 경우 무한 loop를 탈출하는 break를 실행합니다.)

```
}
// 이동하려는 곳이 유효한 좌표인지 체크하는 함수
bool checkXY(int x, int y) {
    return (x >= 0 && x < mapX && y >= 0 && y < mapY);
}
```

4. 테스트

•(테스트 : 입력에 따른 원하는 결과나 나오는지 확인하는 과정)

1. 기능 별 테스트 결과: (요구사항 별 스크린샷)

-지도올 출력하고 벗어나는 경우 경고를 하는 기능

현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): 지도
USER |아이템| 적 | |목적지|

아이템 | | | 적 | | |
적	포션			
포션 | | | | | 적 |

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

현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료):

-상, 하, 좌, 우 입력에 따라 위치를 이동하는 기능(이동 할 때마다 체력이 1씩 마이너스),
종료 입력시

현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): 하
|아이템| 적 | |목적지|

USER | | | 적 | | |
적	포션			
포션 | | | | | 적 |

아이템이 있습니다.

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

USER |아이템| 적 | |목적지|
아이템 | | | 적 | | |
적	포션			
포션 | | | | | 적 |

현재 HP: 18 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
| USER | 적 | |목적지|

아이템 | | | 적 | | |
적	포션			
포션 | | | | | 적 |

아이템이 있습니다.

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

USER |아이템| 적 | |목적지|
아이템 | | | 적 | | |
적	포션			
포션 | | | | | 적 |

현재 HP: 16 명령어를 입력하세요 (상,하,좌,우,지도,종료):

(종료 입력시)

윙프를 기다립니다.
현재 HP: 50 윙프어를 입력하세요 (윙,와,와,승,외,온,윙프): 윙프

-아이템에 따라 체력에 영향을 주는 기능(+안내 메시지 기능)

USER	아이템	적	목적지
아이템		적	
	적	포션	
포션			적

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

USER	적	목적지
아이템		적
	적	포션
포션		적

아이템이 있습니다.

(아이템과 만난 경우)

USER	적	목적지
아이템		적
	적	포션
포션		적

아이템이 있습니다.

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

아이템	USER	목적지
아이템		적
	적	포션
포션		적

적이 있습니다. HP가 2 줄어듭니다.

현재 HP: 12 명령어를 입력하세요 (상,하,좌,우,지도,종료):

(적과 만난 경우)

```

|아이템| 적 | |목적지|
-----
아이템| | |적| |
-----
| | USER | | |
-----
| 적 | 포션 | | |
-----
포션| | | |적|
-----
현재 HP: 10 명령어를 입력하세요 (상,하,좌,우,지도,종료): 하
|아이템| 적 | |목적지|
-----
아이템| | |적| |
-----
| | | | |
-----
| 적 | USER | | |
-----
포션| | | |적|
-----
포션이 있습니다. HP가 2 증가합니다.
현재 HP: 11 명령어를 입력하세요 (상,하,좌,우,지도,종료): 
(포션과 만난 경우)

```

-체력이 0이하가 되면 게임을 종료하는 기능

```

현재 HP: 1 명령어를 입력하세요 (상,하,좌,우,지도,종료): 상
|아이템| USER | |목적지|
-----
아이템| | |적| |
-----
| | | | |
-----
| 적 | 포션 | | |
-----
포션| | | |적|
-----
적이 있습니다. HP가 2 줄어듭니다.
HP가 0 이하가 되었습니다.
게임을 종료합니다.

```

-목적지에 도달하면 게임을 종료하는 기능

```

|아이템| 적 | USER |목적지|
-----
아이템| | |적| |
-----
| | | | |
-----
| 적 | 포션 | | |
-----
포션| | | |적|
-----
현재 HP: 15 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
|아이템| 적 | |USER|
-----
아이템| | |적| |
-----
| | | | |
-----
| 적 | 포션 | | |
-----
포션| | | |적|
-----
목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.

```

2. 최종 테스트 스크린샷: (프로그램 전체 동작 스크린샷)

(최종 플레이 스크린 샷)

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

USER |아이템| 적 | |목적지|

아이템 | | |적 | |

| | | | |

|적 |포션 | | |

포션 | | | |적 |

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

아이템 |적 | |목적지|

USER | | |적 | |

| | | | |

|적 |포션 | | |

포션 | | | |적 |

아이템이 있습니다.

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

아이템 |적 | |목적지|

아이템 | | |적 | |

USER | | | | |

|적 |포션 | | |

포션 | | | |적 |

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

아이템 |적 | |목적지|

아이템 | | |적 | |

| | | | |

USER |적 |포션 | | |

포션 | | | |적 |

현재 HP: 17 명령어를 입력하세요 (상,하,좌,우,지도,종료): 하
|아이템|적|목적지|

아이템| |적| |
| | | |
|적|포션| |
USER| |적|

포션이 있습니다. HP가 2 증가합니다.
현재 HP: 18 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
|아이템|적|목적지|

아이템| |적| |
| | | |
|적|포션| |
포션|USER| |적|

현재 HP: 17 명령어를 입력하세요 (상,하,좌,우,지도,종료): 상
|아이템|적|목적지|

아이템| |적| |
| | | |
|USER|포션| |
포션| |적|

적이 있습니다. HP가 2 줄어듭니다.
현재 HP: 14 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
|아이템|적|목적지|

아이템| |적| |
| | | |
|적|USER| |
포션| |적|

포션이 있습니다. HP가 2 증가합니다.
현재 HP: 15 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
|아이템|적|목적지|

아이템| |적| |
| | | |
|적|포션|USER| |
포션| |적|

현재 HP: 14 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
|아이템|적|목적지|

아이템| |적| |
| | | |
|적|포션|USER| |
포션| |적|

현재 HP: 13 명령어를 입력하세요 (상,하,좌,우,지도,종료): 상
|아이템|적|목적지|

아이템| |적| |
| | | |
|적|포션| |
포션| |적|

현재 HP: 12 명령어를 입력하세요 (상,하,좌,우,지도,종료): 상
|아이템|적|목적지|

아이템| |적|USER| |
| | | |
|적|포션| |
포션| |적|

현재 HP: 11 명령어를 입력하세요 (상,하,좌,우,지도,종료): 상
아이템 | 적 | USER |

아이템 | | 적 | |

| | | | |

| 적 | 포션 | | |

포션 | | | | 적 |

목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.

-승리

현재 HP: 20 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
USER | 적 | 목적지 |

아이템 | | 적 | |

| | | | |

| 적 | 포션 | | |

포션 | | | | 적 |

아이템이 있습니다.

현재 HP: 19 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
아이템 | USER | 목적지 |

아이템 | | 적 | |

| | | | |

| 적 | 포션 | | |

포션 | | | | 적 |

적이 있습니다. HP가 2 줄어듭니다.

현재 HP: 16 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
아이템 | 적 | USER | 목적지 |

아이템 | | 적 | |

| | | | |

| 적 | 포션 | | |

포션 | | | | 적 |

현재 HP: 15 명령어를 입력하세요 (상,하,좌,우,지도,종료): 하
아이템 | 적 | 목적지 |

아이템 | | USER | |

| | | | |

| 적 | 포션 | | |

포션 | | | | 적 |

적이 있습니다. HP가 2 줄어듭니다.

```

현재 HP: 12 명령어를 입력하세요 (상,하,좌,우,지도,종료): 좌
|아이템| 적 | 목적지|
-----
아이템| | USER | 적 | |
-----
| | | | | |
-----
| 적 | 포션 | | | |
-----
포션 | | | | | 적 |
-----
현재 HP: 11 명령어를 입력하세요 (상,하,좌,우,지도,종료): 상
|아이템| USER | 목적지|
-----
아이템| | | 적 | |
-----
| | | | | |
-----
| 적 | 포션 | | | |
-----
포션 | | | | | 적 |
-----
적이 있습니다. HP가 2 줄어듭니다.
현재 HP: 8 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
|아이템| 적 | USER | 목적지|
-----
아이템| | | 적 | |
-----
| | | | | |
-----
| 적 | 포션 | | | |
-----
포션 | | | | | 적 |
-----
현재 HP: 7 명령어를 입력하세요 (상,하,좌,우,지도,종료): 하
|아이템| 적 | 목적지|
-----
아이템| | | USER | |
-----
| | | | | |
-----
| 적 | 포션 | | | |
-----
포션 | | | | | 적 |
-----
적이 있습니다. HP가 2 줄어듭니다.
현재 HP: 4 명령어를 입력하세요 (상,하,좌,우,지도,종료): 좌
|아이템| 적 | 목적지|
-----
아이템| | USER | 적 | |
-----
| | | | | |
-----
| 적 | 포션 | | | |
-----
포션 | | | | | 적 |
-----
현재 HP: 3 명령어를 입력하세요 (상,하,좌,우,지도,종료): 상
|아이템| USER | 목적지|
-----
아이템| | | 적 | |
-----
| | | | | |
-----
| 적 | 포션 | | | |
-----
포션 | | | | | 적 |
-----
적이 있습니다. HP가 2 줄어듭니다.
HP가 0 이하가 되었습니다.
게임을 종료합니다.

```

-패배

5. 결과 및 결론

1. 프로젝트 결과: mud game 베이스 코드에 체력과 아이템 상호작용이라는 두 가지 기능을

추가하고 중복되는 코드를 함수화 하여 게임을 완성시켰다.

2. 느낀 점: 함수화를 할 때 매개변수 설정이 항상 힘들었는데, 이번 활동을 통해 2차원 배열을 매개변수로 지정할 때 1차원의 수가 정해져야 한다는 점과 (map[][mapX]), 어느 경우에 참조변수를 매개변수로 사용해야 되는지 감이 잡히게 되어서 유익했습니다.