

C++ Programming Week 9 Project

Mud game

214935 이승진

1. 서론

1. 프로젝트 목적 및 배경: 7 주차까지 배운 내용에 대한 실습을 위해 진행
2. 목표: 문제에서 요구하는 기능들을 추가한 Mud 게임 구현

2. 요구사항

1. 사용자 요구사항: 유저가 상하좌우로만 이동하며 목적지에 도착하는 게임
지도를 출력하고, 종료할 수 있는 기능 추가
지도 밖으로 나가게 되면 에러, 목적지에 도착하면 성공 메시지 출력
최초 체력(hp) 20을 보유하며, 이동시 1의 체력 소모
명령을 입력 받을 때마다 현재 체력을 표시함
체력이 0이 되면 실패 메시지를 출력하고 종료
아이템(무기/갑옷), 포션, 적을 만났을 때 각 상황에 맞는 메시지 출력

2. 기능 계획

맵 생성 및 출력

- 5x5 맵을 생성하고, 각 위치에 특정 객체(아이템, 적, 포션, 목적지)를 배치
- 사용자의 현재 위치를 포함하여 지도를 출력

사용자 이동 및 상호작용

- 사용자가 상, 하, 좌, 우 방향으로 이동할 수 있도록 함
- 사용자가 이동할 때 체력이 감소하고, 맵 경계를 넘지 않도록 제한

위치 별 상호작용

- 이동 중 특정 위치에서 적을 만나거나 아이템 및 포션을 발견할 수 있다. 적을 만나면 체력이 감소하고, 포션을 만나면 체력이 회복되도록 함

목적지 도달 여부 체크

- 사용자가 목적지에 도달하면 게임이 종료됨

체력 관리 및 게임 종료

- 체력이 0 이하로 떨어지면 게임 종료.
- 종료 또는 목적지 도달 시, 성공 여부에 따라 메시지를 출력하고 게임을 종료한다.

3. 함수 계획

1. displayMap(int map[][mapX], int user_x, int user_y)
 - 맵과 사용자 위치를 출력합니다.
 - 사용자 위치와 맵의 상태를 시각적으로 표현하여 사용자에게 안내합니다.
2. moveUser(int &user_x, int &user_y, int deltaX, int deltaY, int map[][mapX], int &hp)
 - 사용자 위치를 갱신하고 체력을 감소시킵니다.
 - 새로운 위치가 맵 내에 있는지 확인하고, 위치에 따라 상태를 체크합니다.
 - 위치 이동 후 지도를 다시 출력합니다.
3. checkXY(int user_x, int mapX, int user_y, int mapY)

- 사용자의 위치가 맵 범위 내에 있는지 체크합니다.
 - 범위를 벗어났을 경우 경고 메시지를 출력하고 원래 위치로 되돌아가게 합니다.
4. checkGoal(int map[][mapX], int user_x, int user_y)
- 사용자가 목적지에 도달했는지 확인합니다.
 - 도달 시 게임을 종료하고 성공 메시지를 출력합니다.
5. checkState(int map[][mapX], int &hp, int user_x, int user_y)
- 사용자가 도달한 위치의 상태를 확인합니다.
 - 위치 상태에 따라 적을 만나 체력이 감소하거나 포션으로 체력이 회복됩니다.

3. 설계 및 구현

1. 유저가 상하좌우로만 이동하며 목적지에 도착하는 게임

1) 코드 스크린샷

```
if (user_input == "상") {
    cout << "위쪽으로 이동합니다." << endl;
    moveUser(user_x, user_y, 0, -1, map, hp);
} else if (user_input == "하") {
    cout << "아래쪽으로 이동합니다." << endl;
    moveUser(user_x, user_y, 0, 1, map, hp);
} else if (user_input == "좌") {
    cout << "왼쪽으로 이동합니다." << endl;
    moveUser(user_x, user_y, -1, 0, map, hp);
} else if (user_input == "우") {
    cout << "오른쪽으로 이동합니다." << endl;
    moveUser(user_x, user_y, 1, 0, map, hp);
} else if (user_input == "지도") {
    cout << "지도를 출력합니다." << endl;
    displayMap(map, user_x, user_y);
} else if (user_input == "종료") {
    cout << "게임을 종료합니다." << endl;
    break;
} else {
    cout << "잘못된 입력입니다." << endl;
    continue;
}
```

```
void moveUser(int &user_x, int &user_y, int deltaX, int deltaY, int map[][mapX], int &hp) {
    int nextX = user_x + deltaX;
    int nextY = user_y + deltaY;
```

2) 입력 :

user_x, user_y: 현재 유저 위치

deltaX, deltaY: 유저 이동 방향 (상: (0, -1), 하: (0, 1), 좌: (-1, 0), 우: (1, 0))

map: 5x5 맵 배열

hp: 유저 체력

3) 반환 값 : 없음

4) 결과(블록/함수가 종료된 결과)

유저의 위치가 업데이트 되고, 이동할 때 마다 체력이 1 감소된다.

5) 설명(코드 내 작동 순서, 내용 등)

1. 새로운 위치(nextX, nextY) 계산 후 맵 범위를 확인.
2. 범위를 벗어났다면 경고 메시지를 출력, 유저 위치는 업데이트하지 않음.
3. 범위 내 위치일 경우 위치 갱신, 체력 1 감소
4. checkState()를 호출해 위치 상태를 확인해 체력 변동 등 상태를 업데이트
5. 변경된 위치로 지도 출력.

2. 지도를 출력하고 종료할 수 있는 기능 추가

1) 코드 스크린샷

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

2) 입력 :

user_x, user_y: 현재 유저 위치

map: 5x5 맵 배열

3) 반환 값 : 없음

4) 결과(블록/함수가 종료된 결과)

유저 위치와 맵 내 아이템, 적, 포션, 목적지 등이 시각적으로 표시된 상태로 출력됨.

5) 설명(코드 내 작동 순서, 내용 등)

1. 이중 for문을 사용해 맵의 각 위치를 순회

2. 각 위치 상태를 switch 문으로 구분하여 아이템, 적, 포션, 목적지 등의 내용을 출력.

3. 유저 위치를 "USER"로 출력하여 표시함.

3. 지도 밖으로 나가게 되면 에러, 목적지에 도착하면 성공 메시지 출력

1) 코드 스크린샷

```
// 지도 내 유효한 위치인지 체크하는 함수
bool checkXY(int user_x, int mapX, int user_y, int mapY) {
    return (user_x >= 0 && user_x < mapX && user_y >= 0 && user_y < mapY);
}

// 유저가 목적지에 도달했는지 체크하는 함수
bool checkGoal(int map[][mapX], int user_x, int user_y) {
    return (map[user_y][user_x] == 4);
}

if (checkGoal(map, user_x, user_y)) {
    cout << "목적지에 도착했습니다! 축하합니다!" << endl;
    break;
}

if (!checkXY(nextX, mapX, nextY, mapY)) {
    cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
}
```

2) 입력 :

user_x, user_y: 현재 유저 위치

mapX, mapY: 맵의 가로세로 크기

3) 반환 값 :

checkXY() : bool

checkGoal() : bool

4) 결과(블록/함수가 종료된 결과)

유효하지 않은 위치면 에러 메시지 출력, 목적지 도착 시 성공 메시지 출력

5) 설명(코드 내 작동 순서, 내용 등)

1. checkXY() : 유저의 위치 확인 후, 범위 밖이면 경고 메시지 출력

2. checkGoal() : 유저의 위치가 목적지인지 확인, 도착 시 축하 메시지 출력 후 종료

4. 최초 체력(hp)을 20 보유하며, 이동 시 1 의 체력 소모

1) 코드 스크린샷

```
int hp = 20; // 초기 체력

void moveUser(int &user_x, int &user_y, int deltaX, int deltaY, int map[][mapX], int &hp) {
    int nextX = user_x + deltaX;
    int nextY = user_y + deltaY;

    if (!checkXY(nextX, mapX, nextY, mapY)) {
        cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
    } else {
        user_x = nextX;
        user_y = nextY;
        hp--; // 이동할 때마다 체력 1 감소
        checkState(map, hp, user_x, user_y);
        displayMap(map, user_x, user_y);
    }
}
```

2) 입력 :

hp : 초기 체력 값 20

3) 반환 값 : 없음

4) 결과(블록/함수가 종료된 결과)

초기 체력 20 설정 후 이동할 때마다 체력이 1 씩 감소함

5) 설명(코드 내 작동 순서, 내용 등)

1. main() 함수에서 체력을 초기화 (hp = 20)

2. moveUser() 함수 내에서 사용자가 움직일 때마다 hp—로 체력을 1 감소시킨다.

5. 명령을 입력 받을 때마다 현재 체력을 표시

1) 코드 스크린샷

```
while (true) {
    string user_input = "";
    cout << "현재 HP : " << hp << " | 명령어를 입력하세요 (상, 하, 좌, 우, 지도, 종료): ";
    cin >> user_input;
```

2) 입력 :

user_input : 사용자로부터 받은 이동 명령(상,하,좌,우,지도,종료)

3) 반환값 : 없음

4) 결과(블록/함수가 종료된 결과)

입력 명령마다 현재 체력 값이 표시됨

5) 설명(코드 내 작동 순서, 내용 등)

1. while 루프에서 명령을 입력 받기 전 현재 HP : <hp> 형태로 체력을 출력함

2. 각 명령어 입력마다 체력이 표시됨

6. 체력이 0 이 되면 실패 메시지를 출력하고 종료

1) 코드 스크린샷

```
if (hp <= 0) {  
    cout << "체력이 모두 소진되었습니다. 공략에 실패하였습니다." << endl;  
    break;  
}
```

2) 입력 :

hp : 유저 체력 값

3) 반환 값 : 없음

4) 결과(블록/함수가 종료된 결과)

체력이 0 이하가 되면 실패 메시지를 출력하고 게임 종료

5) 설명(코드 내 작동 순서, 내용 등)

1. 이동이나 적을 만나 체력이 감소 할 때마다 $hp \leq 0$ 조건을 충족하는지 검사
2. 조건을 충족하면 메시지를 출력 후 break 로 게임 루프를 종료함

7. 아이템(무기/갑옷), 포션, 적을 만났을 때 각 상황에 맞는 메시지 출력

1) 코드 스크린샷

```
void checkState(int map[][mapX], int &hp, int user_x, int user_y) {  
    switch (map[user_y][user_x]) {  
        case 1:  
            cout << "아이템이 있습니다." << endl;  
            break;  
        case 2:  
            hp -= 2; // 적을 만나면 체력 2 감소  
            cout << "적을 만났습니다! 체력이 2 감소합니다." << endl;  
            break;  
        case 3:  
            hp += 2; // 포션을 만나면 체력 2 증가  
            cout << "포션을 발견했습니다! 체력이 2 회복됩니다." << endl;  
            break;  
    }  
}
```

2) 입력 :

map : 맵 배열

hp : 유저 체력

user_x, user_y : 현재 유저 위치

3) 반환 값 : 없음

4) 결과(블록/함수가 종료된 결과)

유저가 특정 위치에서 아이템/적/포션을 만나 메시지가 출력되고 체력이 증가 또는 감소

5) 설명(코드 내 작동 순서, 내용 등)

1. switch 문을 사용하여 유저의 위치 상태(map[user_y][user_x])를 확인
2. 아이템 위치에 도달하였을 때, “아이템이 있습니다.” 를 출력함
3. 적 위치에 도달하였을 때, 체력 2 감소 및 “적을 만났습니다! 체력이 2 감소합니다.” 출력
4. 포션 위치에 도달하였을 때, 체력 2 증가 및 “포션을 발견했습니다! 체력이 2 회복됩니다.” 출력

4. 테스트

1. 기능 별 테스트 결과: (요구사항 별 스크린샷)

1) 유저가 상하좌우로만 이동하며 목적지에 도착하는 게임

현재 HP : 20 | 명령어를 입력하세요 (상, 하, 좌, 우, 지도, 종료): 하
아래쪽으로 이동합니다.
아이템이 있습니다.

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

2) 지도를 출력하고, 종료할 수 있는 기능 추가

현재 HP : 19 | 명령어를 입력하세요 (상, 하, 좌, 우, 지도, 종료): 지도
지도를 출력합니다.

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

현재 HP : 19 | 명령어를 입력하세요 (상, 하, 좌, 우, 지도, 종료): 종료
게임을 종료합니다.

3) 지도 밖으로 나가게 되면 에러, 목적지에 도착하면 성공 메시지 출력

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

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

| | | | | | | | |
|-------|-------|---|--|-----|--|------|--|
| | 아 이 템 | 적 | | | | USER | |
| ----- | | | | | | | |
| 아 이 템 | | | | 적 | | | |
| ----- | | | | | | | |
| | | | | | | | |
| ----- | | | | | | | |
| | | 적 | | 포 셴 | | | |
| ----- | | | | | | | |
| 포 셴 | | | | | | 적 | |
| ----- | | | | | | | |

목적지에 도착했습니다! 축하합니다!

4) 최초 체력(hp) 20 을 보유하며, 이동시 1 의 체력 소모

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

아이템이 있습니다.

| | | | | | |
|-------|-------|---|--|-----|-------|
| | 아 이 템 | 적 | | | 목 적 지 |
| ----- | | | | | |
| USER | | | | 적 | |
| ----- | | | | | |
| | | | | | |
| ----- | | | | | |
| | | 적 | | 포 셴 | |
| ----- | | | | | |
| 포 셴 | | | | | 적 |
| ----- | | | | | |

5) 명령을 입력 받을 때마다 현재 체력을 표시함

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

아이템이 있습니다.

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

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

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

6) 체력이 0 이 되면 실패 메시지를 출력하고 종료

```
현재 HP : 1 | 명령어를 입력하세요 (상, 하, 좌, 우, 지도, 종료): 우
오른쪽으로 이동합니다.
적을 만났습니다! 체력이 2 감소합니다.
  |아이템| 적 | |목적지|
-----
아이템 |   |   | 적 |   |
-----
      |   |   |   |   |
-----
      | USER | 포션 |   |   |
-----
포션 |   |   |   | 적 |
-----
체력이 모두 소진되었습니다. 공략에 실패하였습니다.
```

7) 아이템(무기/갑옷), 포션, 적을 만났을 때 각 상황에 맞는 메시지 출력

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

현재 HP : 17 | 명령어를 입력하세요 (상, 하, 좌, 우, 지도, 종료): 하
아래쪽으로 이동합니다.
적을 만났습니다! 체력이 2 감소합니다.
  |아이템| 적 | |목적지|
-----
아이템 |   |   | 적 |   |
-----
      |   |   |   |   |
-----
      | USER | 포션 |   |   |
-----
포션 |   |   |   | 적 |
-----

현재 HP : 14 | 명령어를 입력하세요 (상, 하, 좌, 우, 지도, 종료): 우
오른쪽으로 이동합니다.
포션을 발견했습니다! 체력이 2 회복됩니다.
  |아이템| 적 | |목적지|
-----
아이템 |   |   | 적 |   |
-----
      |   |   |   |   |
-----
      | 적 | USER |   |   |
-----
포션 |   |   |   | 적 |
-----
```

2. 최종 테스트 스크린샷: (프로그램 전체 동작 스크린샷)

현재 HP : 20 | 명령어를 입력하세요 (상, 하, 좌, 우, 지도, 종료): 지도
지도를 출력합니다.

USER |아이템| 적 | |목적지|

아이템 | | |적 | |

| | | | |

|적 |포션 | |

포션 | | |적 |

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

|아이템| 적 | |목적지|

아이템 | USER | |적 | |

| | | | |

|적 |포션 | |

포션 | | |적 |

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

|아이템| 적 | |목적지|

아이템 | | |적 | |

| USER | | |

|적 |포션 | |

포션 | | |적 |

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

적을 만났습니다! 체력이 2 감소합니다.

|아이템| 적 | |목적지|

아이템 | | |적 | |

| | | | |

| USER |포션 | |

포션 | | |적 |

현재 HP : 14 | 명령어를 입력하세요 (상, 하, 좌, 우, 지도, 종료): 우
오른쪽으로 이동합니다.

포션을 발견했습니다! 체력이 2 회복됩니다.

|아이템| 적 | |목적지|

아이템 | | |적 | |

| | | | |

|적 |USER | |

포션 | | |적 |

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

|아이템| 적 | |목적지|

아이템 | | |적 | |

| | | | |

|적 | 포션 | USER | |

포션 | | |적 |

현재 HP : 14 | 명령어를 입력하세요 (상, 하, 좌, 우, 지도, 종료): 상
위쪽으로 이동합니다.

|아이템| 적 | |목적지|

아이템 | | |적 | |

| | | USER | |

|적 | 포션 | | |

포션 | | |적 |

현재 HP : 13 | 명령어를 입력하세요 (상, 하, 좌, 우, 지도, 종료): 상
위쪽으로 이동합니다.

적을 만났습니다! 체력이 2 감소합니다.

|아이템| 적 | |목적지|

아이템 | | |USER | |

| | | | |

|적 | 포션 | | |

포션 | | |적 |

현재 HP : 10 | 명령어를 입력하세요 (상, 하, 좌, 우, 지도, 종료): 우
오른쪽으로 이동합니다.

|아이템| 적 | |목적지|

아이템 | | |적 | USER |

| | | | |

|적 | 포션 | | |

포션 | | |적 |

현재 HP : 9 | 명령어를 입력하세요 (상, 하, 좌, 우, 지도, 종료): 상
위쪽으로 이동합니다.

|아이템| 적 | |USER |

아이템 | | |적 | |

| | | | |

|적 | 포션 | | |

포션 | | |적 |

목적지에 도착했습니다! 축하합니다!

5. 결과 및 결론

1. 프로젝트 결과

맵 탐색 및 이동, 목적지 도달 체크, 이벤트 발생 및 상태 확인, 체력 관리 및 게임 종료, 상,하,좌,우,지도,종료 명령을 이용한 유저 인터페이스 기능이 있는 텍스트 기반으로 플레이하는 mud game 을 제작하였다.

2. 느낀 점

각 기능들을 함수를 사용해 분리하여 작성하니 가독성이 높아지고, 코드를 쉽게 변경할 수 있었고, 함수 표현의 중요성을 깨닫게 되었다. 또한 평소에 관심있던 게임 프로그래밍에 대한 기본적인 로직과 사용자 인터페이스 설계에 대해 배울 수 있었으며, 확장 가능한 코드를 설계하고, 유저에게 직관적인 피드백을 주는 것이 중요하다는 것을 느낄 수 있었다.