

서론

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

요구사항

- 사용자 요구사항: 유저가 상하좌우로만 이동하며 목적지에 도착하는 게임
- 기능 계획
 - 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
 - 지도 밖으로 나가게 되면 에러 메시지 출력
 - 목적지에 도착하면 “성공”을 출력하고 종료
- 추가 기능 계획
 - 유저는 체력 20을 가지고 게임 시작
 - 사용자가 이동할 때 마다 사용자 체력 1씩 감소
 - 처음 명령문을 입력 받을 때 마다 HP 함께 출력
 - HP가 0이 되면 “실패”를 출력하고 종료
 - 무기/갑옷, 포션, 적을 만났을 때 그에 대한 메시지를 출력
- 함수 계획
 - 맵 유효성 함수: checkXY()
 - 맵과 유저 위치를 표시하는 함수: displayMap()
 - 목적지에 도달 체크 함수: checkGoal()
 - 유저 아이템 함수: checkState()

설계 및 구현

```
// 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}
    };
    //좌표, 체력, 무기, 갑옷
    int user_x = 0;
    int user_y = 0;
    int hp = 20;
    bool weapon = false;
    int armor = 0;
```

설명

- map[mapY][mapX]: 5x5 크기의 2차원 배열로 맵을 구성, 배열의 각 요소는 특정 값을 가지며, 값에 따라 위치의 상태를 나타냄
- hp: 플레이어의 초기 체력, 기본값 20
- weapon: 플레이어가 무기를 가지고 있는지 여부를 나타내는 변수
- armor: 플레이어가 가지고 있는 갑옷의 개수

결과

- map 배열과 초기 변수들이 설정된 후, 게임이 시작되고 플레이어의 위치와 상태가 초기화

```
string user_input; // 사용자의 입력을 저장할 변수
cout << "명령어를 입력하세요 (up,down,left,right,map,exit): ";
cin >> user_input;

if (user_input == "up") {
    // 위로 한 칸 올라가기
    user_y -= 1;
    if (!checkXY(user_x, user_y, mapX, mapY)) {
        cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
        user_y += 1;
    } else {
        hp -= 1; //체력 -
        cout << "위로 한 칸 올라갑니다. 남은 HP: " << hp << endl;
        displayMap(map, user_x, user_y);
    }
}
```

입력

- 사용자가 "up", "down", "left", "right", "map", "exit" 중 하나를 입력
- user_x, user_y: 플레이어의 현재 위치 좌표
- hp: 플레이어의 체력으로, 이동 시마다 1씩 감소

결과

- up 명령어를 입력하면 user_y 값이 1 감소하여 위로 이동
- 위치가 유효하지 않으면 "맵을 벗어났습니다. 다시 돌아갑니다." 메시지를 출력, 위치가 유효하지 않으면 "맵을 벗어났습니다. 다시 돌아갑니다." 메시지를 출력
- 위치가 유효하지 않으면 "맵을 벗어났습니다. 다시 돌아갑니다." 메시지를 출력
- 현재 맵과 플레이어 위치를 displayMap 함수를 통해 출력

```
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 |"; // 유저 위치 표시
            } else {
                // 해당 위치의 상태 출력
                switch (map[i][j]) {
                    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;
    }
}
```

입력

- map: 5x5 크기의 2차원 배열로, 각 위치의 상태 표시
- user_x: 플레이어의 현재 x 좌표
- user_y: 플레이어의 현재 y 좌표

결과

- 현재 맵의 각 위치와 플레이어의 위치를 콘솔에 출력
- 플레이어의 위치는 "USER"로 표시

설명

- 이중 for 루프를 통해 map 배열의 모든 위치를 탐색
- 현재 위치가 플레이어의 위치(user_x, user_y)와 일치하면 "USER"를 출력하여 플레이어의 현재 위치를 표시
- 그렇지 않은 경우 map[i][j]의 값에 따라 switch 문을 사용해 해당 위치의 상태를 출력
- 각 줄의 끝에는 cout << endl을 통해 줄바꿈을 하며, 가로 구분선 출력

```
// 이동하려는 좌표 유효 체크
bool checkXY(int user_x, int user_y, int mapX, int mapY) {
    return (user_x >= 0 && user_x < mapX && user_y >= 0 && user_y < mapY);
}
```

입력

- user_x: 플레이어의 현재 x 좌표
- user_y: 플레이어의 현재 y 좌표
- mapX: 맵의 x축 크기
- mapY: 맵의 y축 크기

반환값

- bool: 입력된 좌표가 맵 내에 있는지 여부를 반환
- true: 좌표가 맵 범위 내에 있음.
- false: 좌표가 맵 범위를 벗어남.

설명

- 맵의 유효 범위 내에 있는지 확인
- user_x가 0 이상 mapX 미만인지, 그리고 user_y가 0 이상 mapY 미만인지 조건을 통해 확인
- 조건이 모두 참이면 true를 반환하여 좌표가 유효하다고 판단, 하나라도 벗어나면 false를 반환

```
// 유저 목적지 도달 체크
bool checkGoal(int map[][mapX], int user_x, int user_y) {
    return map[user_y][user_x] == 4;
}
```

반환값

- bool: 플레이어가 목적지에 도달했는지 여부를 반환
- true: 현재 위치가 목적지임.
- false: 현재 위치가 목적지가 아님.

설명

- 이 함수는 플레이어의 현재 위치(user_x, user_y)가 map 배열에서 4인지 확인
- map[user_y][user_x] == 4 조건을 통해, 현재 위치가 목적지일 경우 true를 반환, 그렇지 않으면 false를 반환
- 이 함수는 게임 진행 중 플레이어가 목적지에 도달했는지 확인하는 데 사용

```
// 유저 상태 체크 (아이템, 적, 포션)
void userItem(int map[][mapX], int user_x, int user_y, int& hp, bool& weapon, int& armor) {
    switch (map[user_y][user_x]) {
        case 1:
            if (rand() % 2 == 0) {
                weapon = true;
                cout << "무기 획득!" << endl;
            } else {
                armor++;
                cout << "갑옷 획득! 현재 갑옷 개수: " << armor << endl;
            }
            break;
        case 2:
            cout << "적을 만났습니다! ";
            if (armor > 0) {
                armor--;
                cout << "갑옷으로 적의 공격을 막았습니다. 남은 갑옷: " << armor << endl;
            } else if (weapon) {
                hp -= 1;
                cout << "무기로 적을 베었습니다. HP -1. 현재 HP: " << hp << endl;
            }
            break;
        case 3:
            hp += 2;
            cout << "포션 HP +2. 현재 HP: " << hp << endl;
            break;
    }
}
```

입력

- map: 2차원 배열로, 각 위치의 상태(빈 공간, 아이템, 적, 포션, 목적지)
- user_x: 플레이어의 현재 x 좌표
- user_y: 플레이어의 현재 y 좌표
- weapon: 플레이어의 무기 소지 여부를 나타내는 값
- armor: 플레이어가 가지고 있는 갑옷의 개수

결과

- 현재 위치의 맵 상태에 따라 플레이어의 아이템 소지 상태, 체력, 갑옷 개수 업데이트
- 상태에 따라 출력 메시지를 제공
 - case 1: 아이템을 획득할 확률 50%, 무기를 획득하면 weapon이 true로 설정, 그렇지 않으면 armor가 1 증가
 - case 2: armor가 있을경우 hp가 1 감소하여 적의 공격을 방어, weapon이 있으면 hp가 1 감소하여 적을 공격.
 - case 3: 포션을 획득하여 hp가 2 증가

설명

- map[user_y][user_x] 값을 기반으로 플레이어의 상태를 업데이트
- case 1 (아이템): rand() % 2 == 0 조건으로 무기와 갑옷 중 하나를 무작위로 획득
- case 2 (적): 플레이어가 적을 만나면 우선 갑옷을 사용해 공격을 막고, 갑옷이 없을 경우 무기로 적을 공격하여 체력이 1 감소
- case 3 (포션): 포션을 획득하여 체력(hp)이 2 증가

테스트

- 지도 밖으로 나가게 되면 에러 메시지 출력

```
명령어를 입력하세요 (up,down,left,right,map,exit): up
맵을 벗어났습니다. 다시 돌아갑니다.
명령어를 입력하세요 (up,down,left,right,map,exit): left
맵을 벗어났습니다. 다시 돌아갑니다.
```

- “map”을 입력하면 전체 지도와 함께 현재 위치를 출력

```
명령어를 입력하세요 (up,down,left,right,map,exit): map
USER |아이템|  적  |      |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션  |      |      |      |  적  |
-----
```

- 이동 명령 시 이동 후 hp 감소, 아이템 획득, 맵 출력

```
명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다. 남은 HP: 19
  | USER | 적 | | 목적지 |
-----
아이템 | | | 적 | |
-----
  | | | | |
-----
  | 적 | 포션 | | |
-----
포션 | | | | 적 |
-----
갑옷 획득! 현재 갑옷 개수: 1
```



- 포션 획득시 hp 증가

```
명령어를 입력하세요 (up,down,left,right,map,exit): down
아래로 한 칸 내려갑니다. 남은 HP: 15
  |아이템| 적 | | 목적지|
-----
아이템 | | | 적 | |
-----
  | | | | |
-----
  | 적 | USER | | |
-----
포션 | | | | 적 |
-----
포션 HP +2. 현재 HP: 17
```


- 적을 만났을 때 hp 감소

```

명령어를 입력하세요 (up,down,left,right,map,exit): down
아래로 한 칸 내려갑니다. 남은 HP: 14
      |아이템|  적  |      |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      | USER | 포션 |      |      |
-----
포션  |      |      |      |  적  |
-----
적을 만났습니다! 적에게 공격당했습니다. HP -2. 현재 HP: 12
  
```

- 체력이 0 이하일때 사망, 게임 종료

```

명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다. 남은 HP: 2
      |아이템|  적  |      |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션  |      |      |      | USER |
-----
적을 만났습니다! 적에게 공격당했습니다. HP -2. 현재 HP: 0
체력이 0 이하입니다. 게임이 종료됩니다.
  
```

- 목적지 도착, 게임 종료


```

명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다. 남은 HP: 16
  |아이템|  적  |      | USER |
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
목적지에 도착했습니다! 축하합니다!

```

최종 테스트 스크린샷

```

명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다. 남은 HP: 19
  | USER |  적  | |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
감옥 획득! 현재 감옥 개수: 1
명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다. 남은 HP: 18
  |아이템| USER |      | |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
적을 만났습니다! 감옥으로 적의 공격을 막았습니다. 남은 감옥: 0
명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다. 남은 HP: 17
  |아이템|  적  | USER | |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다. 남은 HP: 16
  |아이템|  적  |      | USER |
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
목적지에 도착했습니다! 축하합니다!

```

결과 및 결론

- 프로젝트 결과: mud game을 만들었다.
- 느낀점: 어째서인지 한글로 상, 하, 좌, 우, 맵, 종료를 입력하면 잘못된 입력이라는 오류가 납니다. 온라인 컴파일러를 이용하면 또 정상적으로 작동하는데... 아직도 이유를 모르겠습니다.

//추가

맵 밖으로 나갔다가 돌아올 때 이벤트가 중복 발생하는 오류가 있어
checkState 함수에 int prev_x, int prev_y를 추가하였습니다

전

```

명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다. 남은 HP: 19
| USER | 적 | | 목적지 |
-----
아이템 | | | 적 | |
-----
| | | | | |
-----
| 적 | 포션 | | | |
-----
포션 | | | | 적 | |
-----

갑옷 획득! 현재 갑옷 개수: 1
명령어를 입력하세요 (up,down,left,right,map,exit): up
맵을 벗어났습니다. 다시 돌아옵니다.
갑옷 획득! 현재 갑옷 개수: 2

```

후

```

명령어를 입력하세요 (up,down,left,right,map,exit): right
오른쪽으로 이동합니다. 남은 HP: 17
|아이템| 적 | |목적지|
-----
아이템| | USER | 적 | |
-----
| | | | |
-----
| 적 | 포션 | | |
-----
포션 | | | | 적 |
-----
명령어를 입력하세요 (up,down,left,right,map,exit): up
위로 한 칸 올라갑니다. 남은 HP: 16
|아이템| USER | |목적지|
-----
아이템| | | 적 | |
-----
| | | | |
-----
| 적 | 포션 | | |
-----
포션 | | | | 적 |
-----
적을 만났습니다! 감옥으로 적의 공격을 막았습니다. 남은 감옥: 0
명령어를 입력하세요 (up,down,left,right,map,exit): up
맵을 벗어났습니다. 다시 돌아갑니다.

```

입력

- prev_x: 플레이어의 이전 x 좌표 (이전 위치 확인용)
- prev_y: 플레이어의 이전 y 좌표 (이전 위치 확인용)

설명

- 현재 좌표 (user_x, user_y)가 이전 좌표 (prev_x, prev_y)와 다른 경우에만 checkState 함수를 통해 상태 업데이트 실행