

[C++프로그래밍및실습] MUD 게임 프로젝트

지역바이오시스템공학과 생물산업기계공학전공 184128 박지환

1. 서론

1-1. 프로젝트 목적 및 배경

이 프로젝트는 간단한 MUD(Multi-User Dungeon) 게임의 개발을 목표로 합니다.

MUD 게임은 주로 텍스트 기반의 어드벤처 게임으로, 플레이어가 특정 공간을 탐험하며 장애물과 아이템을 만나고 목표 지점에 도달하는 게임 유형입니다.

이 프로젝트는 C++를 활용하여 기본적인 게임 로직과 맵 탐색 기능을 구현하고, 콘솔을 통해 사용자와 상호작용할 수 있도록 구성하였습니다.

1-2. 목표

본 프로젝트의 목표는 다음과 같습니다.

- C++의 기초 문법과 배열, 조건문, 반복문, 함수 등을 활용하여 MUD 게임의 로직을 구현합니다.
- 사용자 입력을 통해 게임 캐릭터를 상하좌우로 이동시키고, 아이템 및 장애물을 만나도록 합니다.
- 목적지에 도착하거나, HP가 0이 되면 게임을 종료합니다.

2. 요구사항

2-1. 사용자 요구사항

사용자 요구사항은 다음과 같습니다.

- 사용자는 "상", "하", "좌", "우", "지도", "종료" 중 하나의 명령어를 입력해 캐릭터를 조작할 수 있어야 합니다.
- 맵을 벗어나는 경우 오류 메시지를 출력하고 다시 이전 위치로 돌아와야 합니다.
- 체력(HP)은 20으로 시작하고, 이동 시 체력이 1씩 감소해야 합니다.
- 아이템, 포션, 적을 만나면 그에 따른 메시지를 출력하고, 그에 따른 HP에 변화가 있어야 합니다.
- 체력이 0이 되면 실패 메시지를 출력하고 종료합니다.
- 목적지에 도착하면 성공 메시지를 출력하고 종료합니다.

2-2. 기능 계획

1. 명령어 입력 및 처리: 사용자의 입력을 통해 상하좌우로 이동하거나, 지도를 출력하거나, 게임을 종료합니다.
2. 지도 출력: 현재 위치와 맵 상태를 표시하는 함수를 통해 지도를 표시합니다.
3. 체력 감소: 이동할 때 마다 체력이 감소하고, 체력이 0이 되면 게임을 종료합니다.
4. 아이템 및 장애물 상호작용: 특정 위치에 아이템, 포션, 적이 있으며, 캐릭터가 만났을 때 체력 변화 및 메시지를 출력합니다.

2-3. 함수 계획

1. `main` : 사용자의 입력을 받아 명령어를 처리하고, 각 기능을 수행하는 함수를 호출합니다.
2. `displayMap` : 맵과 현재 위치를 출력합니다.
3. `checkXY` : 현재 위치가 유효한 범위인지 확인합니다.
4. `checkGoal` : 사용자가 목적지에 도착했는지 확인합니다.
5. `checkState` : 사용자가 특정 위치에서 만나는 아이템, 포션, 적에 대한 메시지를 출력하고 체력을 조정합니다.

3. 설계 및 구현

3-1. 기능 별 구현 사항

기능 1. 사용자 명령어 입력 및 처리

```
while (true)
{
    cout << "현재 HP: " << user_hp << endl;
    string user_input = "";

    cout << "명령어를 입력하세요 (상,하,좌,우,지도,종료): ";
    cin >> user_input;

    if (user_input == "상")
    {
        user_y--;
        cout << "위로 한 칸 올라갑니다." << endl;
    }
    else if (user_input == "하")
    {
        user_y++;
        cout << "아래로 한 칸 내려갑니다." << endl;
    }
    else if (user_input == "좌")
    {
        user_x--;
        cout << "왼쪽으로 이동합니다." << endl;
    }
    else if (user_input == "우")
    {
        user_x++;
        cout << "오른쪽으로 이동합니다." << endl;
    }
    else if (user_input == "지도")
    {
        displayMap(map, user_x, user_y);
        continue;
    }
    else if (user_input == "종료")
    {
        break;
    }
    else
    {
        cout << "잘못된 입력입니다." << endl;
        continue;
    }
}
```

- 설명: `main` 함수에서 `cin`을 통해 명령어를 입력받고, 입력된 값에 따라 캐릭터를 이동하거나 특정 기능을 수행합니다.

기능 2. 지도 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 |";
            else
            {
                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 << "\n ----- " << endl;
    }
}
```

- 설명: `displayMap` 함수는 5X5 맵 배열을 출력하고, 현재 캐릭터의 위치에 `USER`를 표시합니다.
- 입력 및 반환값: 맵 배열과 사용자의 위치가 입력됩니다.

기능 3. 위치 유효성 검사

```
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;
}

if (!checkXY(user_x, mapX, user_y, mapY))
{
    cout << "맵을 벗어났습니다. 다시 돌아갑니다." << endl;
    if (user_input == "상")
        user_y++;
    else if (user_input == "하")
        user_y--;
    else if (user_input == "좌")
        user_x++;
    else if (user_input == "우")
        user_x--;
    continue;
}
```

- **설명:** checkXY 함수는 이동 후 위치가 유효한지 확인하며, 맵을 벗어난 경우 경고 메시지를 출력합니다.
- **입력 및 반환값:** 사용자 위치를 입력으로 받아 유효 여부를 반환합니다.

기능 4. 목적지 도착 검사

```
bool checkGoal(int map[][mapX], int user_x, int user_y)
{
    return map[user_y][user_x] == 4;
}
```

- **설명:** checkGoal 함수는 현재 위치가 목적지인지 확인하고, 도착 시 게임을 종료합니다.
- **입력 및 반환값:** 현재 맵과 위치를 입력으로 받아 도착 여부를 반환합니다.

기능 5. 아이템 및 장애물 상호작용

```
void checkState(int map[][mapX], int &user_hp, int user_x, int user_y)
{
    switch (map[user_y][user_x])
    {
        case 1:
            cout << "아이템이 있습니다." << endl;
            break;
        case 2:
            cout << "적이 있습니다. HP가 2 감소합니다." << endl;
            user_hp -= 2;
            if (user_hp <= 0)
            {
                cout << "HP가 0 이하가 되었습니다. 실패했습니다." << endl;
                cout << "게임을 종료합니다." << endl;
                exit(0);
            }
            break;
        case 3:
            cout << "포션을 발견했습니다. HP가 2 회복됩니다." << endl;
            user_hp += 2;
            break;
        case 4:
            cout << "목적지에 도착했습니다!" << endl;
            break;
    }
}
```

- 설명: checkState 함수는 특정 위치에서 아이템, 포션, 적을 만나게 되면 그에 따른 메시지를 출력하고 체력을 조정합니다.
- 입력 및 반환값: 현재 맵, 체력, 사용자 위치를 입력으로 받아 체력 변화와 메시지를 출력합니다.

4. 테스트

4-1. 기능 별 테스트 결과

테스트 1. 상하좌우 이동 시 체력 감소

현재 HP: 20

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

	USER	적		목적지
아이템			적	
	적	포션		
포션				적

아이템이 있습니다.

현재 HP: 19

명령어를 입력하세요 (상, 하, 좌, 우, 지도, 종료): 좌
왼쪽으로 이동합니다.

USER	아이템	적		목적지
아이템			적	
	적	포션		
포션				적

현재 HP: 18

- 입력: "상", "하", "좌", "우"의 명령어를 입력합니다.
- 결과: 체력이 1씩 감소, 각 방향으로 이동 등의 메시지를 출력합니다.

테스트 2. 맵 경계 체크

```
명령어를 입력하세요 (상,하,좌,우,지도,종료): 좌
왼쪽으로 이동합니다.
맵을 벗어났습니다. 다시 돌아갑니다.
현재 HP: 16
명령어를 입력하세요 (상,하,좌,우,지도,종료): 상
위로 한 칸 올라갑니다.
맵을 벗어났습니다. 다시 돌아갑니다.
현재 HP: 16
```

- 입력: 맵 외부로 이동하는 명령어를 입력합니다.
- 결과: 경고 메시지를 출력하고 제자리로 복귀합니다.

테스트 3. 목적지 도착

```
명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 이동합니다.
      |아이템|적|      |USER|
-----
아이템|      |      |적|      |
-----
      |      |      |      |      |
-----
      |적|포션|      |      |
-----
포션|      |      |      |적|
-----
목적지에 도착했습니다!
```

- 입력: 목적지로 이동합니다.
- 결과: 성공 메시지와 함께 게임을 종료합니다.

4-2. 최종 테스트 스크린샷

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

	USER	적		목적지
아이템			적	
	적	포션		
포션				적

아이템이 있습니다.

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

	USER	아이템	적		목적지
아이템				적	
		적	포션		
포션					적

현재 HP: 18
명령어를 입력하세요 (상,하,좌,우,지도,종료): 하
아래로 한 칸 내려갑니다.

		아이템	적		목적지
USER				적	
		적	포션		
포션					적

아이템이 있습니다.

현재 HP: 17
명령어를 입력하세요 (상,하,좌,우,지도,종료): 상
위로 한 칸 올라갑니다.

	USER	아이템	적		목적지
아이템				적	
		적	포션		
포션					적

현재 HP: 16
명령어를 입력하세요 (상,하,좌,우,지도,종료): 좌
왼쪽으로 이동합니다.
맵을 벗어났습니다. 다시 돌아갑니다.

현재 HP: 16
명령어를 입력하세요 (상,하,좌,우,지도,종료): 상
위로 한 칸 올라갑니다.
맵을 벗어났습니다. 다시 돌아갑니다.

현재 HP: 16

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

	USER	적		목적지
아이템			적	
	적	포션		
포션				적

아이템이 있습니다.

현재 HP: 15

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

	아이템	USER		목적지
아이템			적	
	적	포션		
포션				적

적이 있습니다. HP가 2 감소합니다.

현재 HP: 12

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

	아이템	적	USER	목적지
아이템			적	
	적	포션		
포션				적

현재 HP: 11

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

	아이템	적		USER
아이템			적	
	적	포션		
포션				적

목적지에 도착했습니다!

5. 결과 및 결론

5-1. 프로젝트 결과

간단한 콘솔 기반의 MUD 게임을 성공적으로 구현해볼 수 있었습니다.

사용자 명령어에 따라 맵 탐색과 상호작용이 이루어지며, 체력과 상태 변화 기능이 정상적으로 작동합니다.

5-2. 느낀 점

본 프로젝트를 통해 C++의 배열, 조건문, 함수 호출 등을 활용하여 게임 로직을 설계하고 구현하는 능력을 기를 수 있었습니다.

제한된 조건 내에서 흥미로운 게임을 만들기 위해 다양한 아이디어와 코드 최적화가 필요함을 깨닫게 되었습니다.