1. 서론

1) **프로젝트 목적 및 배경 :** 7주차까지 배운 내용에 대한 실습을 위한 진행

2. 요구사항

1) 사용자 요구사항: 상, 하, 좌, 우를 입력하여 받은 후 아이템, 포션 등을 활용하여 적을 피해 목적지까지 이동.

2) 기능 계획

- ①실행에 필요한 변수 선언
- ②입력 받을 명령어 출력 (up, down, left, right, map, exit) 및 입력 받기
- ③움직이는 명령어(up, down, left, right)일 때 좌표 유효성 검사
- ④ 플레이어 좌표 변경 및 플레이어 HP 감소
- ⑤아이템, 적, 포션 상호작용
- ⑥지도 출력
- ⑦HP 출력, 플레이어 HP가 0이면 게임 종료
- ⑧ map, exit 입력 시 각각 지도 출력 및 게임 종료
- ⑨명령어 잘못 입력 시 재입력
- ⑩목적지 도착 시 게임 승리 문구 출력 후 종료

3) 함수 계획

- ① 좌표 유효성 검사
- ② 지도 출력
- ③ 목적지 도착 검사
- ④ 움직임 검사

3. 설계 및 구현

- 1) 기능 별 구현 사항
- ① 실행에 필요한 변수 선언

```
const int mapX = 5;
const int mapY = 5;
int playerHp = 20;
```

전역변수로 플레이어 체력과 맵 크기를 선언

2차원 배열에 적과 아이템 등을 배치할 지도 변수, 플레이어의 현재 위치 인덱스 변수 선언

② 입력 받을 명령어 출력 (up, down, left, right, map, exit) 및 입력받기

```
// 게임 시작
while (1) { // 사용자에게 계속 입력받기 위해 무한 루프
    // 사용자의 입력을 저장할 변수
    string user_input = "";
    //한글 입력시 이상하게 인식해서 영어로 바꿈.
    cout << "명령어를 입력하세요 (up,down,left,right,map,exit): ";
    cin >> user_input;
```

먼저 입력을 계속 받기 위한 무한 루프 속에 user_input 변수 선언 및 입력 받기

③ 움직이는 명령어(up, down, left, right)일 때 좌표 유효성 검사

```
//명령어 확인 후 움직였다면 true반환

bool checkMove(string ch){
    return (ch == "up" || ch == "down" || ch == "left" || ch == "right");
}
```

받은 명령어가 이동명령어(up, down, left, right)라면 true를 반환하는 함수 선언

```
// 이동하려는 곳이 유효한 좌표인지 체크하는 함수 (매개변수 추가 - 유저 명령어)
bool checkYY(int user_x, int mapX, int user_y, int mapY, string user_input) {
    bool checkFlag = false;
    if (user_input == "up")
        user_y--;
    else if (user_input == "down")
        user_y++;
    else if (user_input == "left")
        user_x--;
    else if (user_input == "right")
        user_x++;
    if (user_x >= 0 && user_x < mapX && user_y >= 0 && user_y < mapY) {
        checkFlag = true;
    }
    return checkFlag;
}
```

받은 명령어와 지도, 플레이어 위치를 매개변수로 받아 맵을 벗어나는지에 대한 유효성 검사. 맵을 벗어났다면 false, 유효한 위치라면 true 반환

움직임 상태 검사 후 true라면 이동 했다는 것이므로 플레이어 위치 유효성 검사 함수 호출. 유효하지 않은 위치라면 문구 출력 후 while문 처음으로 이동

④ 플레이어 좌표 변경 및 플레이어 HP 감소

```
//움직였는데 XY가 유효하다면 실행
else{
    playerHp--;
    if (user_input == "up") {
        cout << "위로 한 칸 올라갑니다." << endl;
        user_y --;
    }
    else if (user_input == "down") {
        cout << "아래로 한 칸 내려갑니다." << endl;
        user_y ++;
    }
    else if (user_input == "left") {
        cout << "왼쪽으로 이동합니다." << endl;
        user_x --;
    }
    else if (user_input == "right") {
        cout << "오른쪽으로 이동합니다." << endl;
        user_x ++;
    }
```

위 else블록은 checkMove, checkXY가 모두 true일 때 성공적으로 이동했으므로 실행하는 블록. 이동했으므로 playerHP 1감소. 유저의 명령어에 따라 유저 위치 인덱스 변환.

⑤ 아이템, 적, 포션 상호작용

```
//움직인 칸에 있는 상호제 확인

if (map[user_y][user_x] == 1){
	cout << "아이템을 먹었습니다. " << endl;
}

if (map[user_y][user_x] == 2){
	playerHp-= 2;
	cout << "적을 만났습니다. HP가 추가로 2 감소합니다." << endl;
}

if (map[user_y][user_x] == 3){
	playerHp += 2;
	cout << "포션을 먹었습니다. HP를 2 회복합니다. " << endl;
}
```

이동한 위치에 아이템이 있다면 문구 출력. 적이 있다면 playerHP 2감소, 포션이 있다면 playerHP 2증가.

⑥지도 출력

```
displayMap(map, user_x, user_y);
```

위 상호작용 후 지도를 출력

⑦ HP 출력, 플레이어 HP가 0이면 게임 종료

```
cout << "HP: " << playerHp << endl;
if (playerHp <= 0){
    cout << "HP가 0이하가 되었습니다. 게임을 종료합니다." << endl;
    break;
}
}
```

HP 출력 후 HP가 0보다 작다면 while문 탈출 후 종료

⑧ map, exit 입력 시 각각 지도 출력 및 게임 종료

```
//움직이지 않았다면 실행
else if (user_input == "map") {
    // TODO: 지도 보여주기 함수 호출
    displayMap(map, user_x, user_y);
}
else if (user_input == "exit") {
    cout << "종료합니다.";
    break;
}
```

명령어가 map이라면 지도 보여주는 함수 호출, exit라면 문구 출력 후 while문 탈출

⑨ 명령어 잘못 입력 시 재입력

```
else{
    cout << "잘못된 입력입니다. : " << user_input << endl;
    continue;
}
```

명령어 리스트에 없는 명령어를 친다면 while문 처음으로 다시 리턴.

⑩ 목적지 도착 시 게임 승리 문구 출력 후 종료

```
// 유저의 위치가 목적지인지 체크하는 함수
bool checkGoal(int map[][mapX], int user_x, int user_y) {
    // 목적지 도착하면
    if (map[user_y][user_x] == 4) {
        return true;
    }
    return false;
}
```

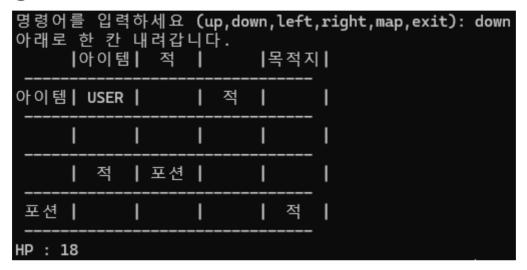
유저의 인덱스의 위치에 목적지인 4가 있다면 true를 반환하는 checkGoal함수 선언

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

checkGoal이 true라면(목적지에 도착했다면) 문구 출력후 while문 탈출

4. 테스트

- 1) 기능 별 테스트 결과
- ② 입력 받을 명령어 출력 (up, down, left, right, map, exit) 및 입력받기 명령어를 입력하세요 (up,down,left,right,map,exit):
- ④ 플레이어 좌표 변경 및 플레이어 HP 감소



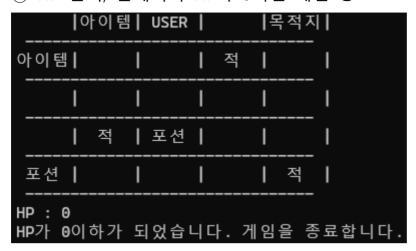
⑤ 아이템, 적, 포션 상호작용

위로	한 템을	입력 칸 올 먹었 USER	라 습	·갑니[·니다.	다. ·				,right,map,exit):	цр
아이	 템		I		I	 적	ı		Ī	
	ı		I		I		ı		1	
	ı	 적	I	포션	I		I		1	
 포션 	I		I		I		I	적 	T	

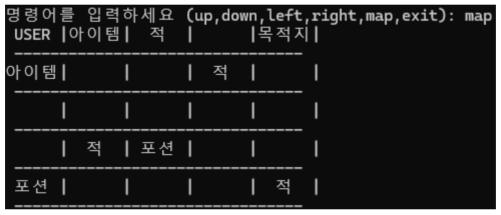
오른	목	으 만닜	로 0 습니	동 디	합니	다. 가	추가	로	2 감	,right,map,exit): ri 소합니다. 	ght
아이	템	I		I		I	 적	ı		T	
		I		I		I		ı		1	
		I	 적	I	포션	I		ı		1	
프-	 년	I		I		I		ı	 적	1	
HP :	: 1	4									

HΡ	:	12	2												
명	령 (거		입력	하	세요	(u	p, do	wn ,	lef1	t,righ	nt,map	o, ex	it):	down
						려갑니									
포	션실	을	먹	었습	ű	다. +	ιΡ를	2	회본	함	니다.				
		١	아	이템	ľ	적	I		=	적	지ㅣ				
아	이 팀	 템			I		I	 적	I		ı				
_															
		١			ı		1		- 1		- 1				
_				·	1	USER	1				- ,				
_				~i 	<u> </u>	U3EK	<u> </u>				_ '				
<u> </u>	 [셔				ī		1			 적	1				
_	- Ľ	'			<u>'</u>		<u>'</u>				- '				
HP	:	13	3												
• • • •	•														

⑦ HP 출력, 플레이어 HP가 0이면 게임 종료



⑧ map, exit 입력 시 각각 지도 출력 및 게임 종료

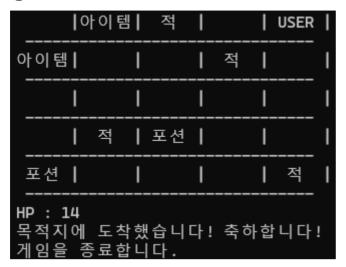


명령어를 입력하세요 (up,down,left,right,map,exit): exit 종료합니다.

⑨ 명령어 잘못 입력 시 재입력

명령어를 입력하세요 (up,down,left,right,map,exit): what 잘못된 입력입니다. : what

⑩ 목적지 도착 시 게임 승리 문구 출력 후 종료



2) 최종 테스트 스크린샷: (프로그램 전체 동작 스크린샷)

PS C:\ USER				ang\0i 적	ne[rive		분 호 목적지	ㅏ면\CPP2409\project\week9> ./9_1_2_mud_game.exe
아이템	1		Ī		Ī	적	Ī		1
	I		١		I		Ī		Ì
Carlottelade	1	적	١	포션	١	destrolestes	1		1
 포션	Ī		1		١		1	 적	1
잘못된	메이 이 메마 OI 에비	[력입 입력 로 이	니 하 동 습	다. 세요 합니 니다	: W (i 다.	hat p,do	wn		right,map,exit): what right,map,exit): right
 아이템	Ī		ī		1	적	Ī		Ţ
	Ī		Ī		Ī		Ī		Ĭ
	1	적	Ī	포션	Ī		Ī		1
 포션	I		1		ī		1	 적	1
오른쪽	를 으 만 났	로 0 , 습니	동 다	합니	다. 가	추가	로		,right,map,exit): right 소합니다.
아이템	I		1		١	 적	1		1
	1		1		1	~~~	Ī		Ţ
	Ī	적	Ī	포션	Ī		Ī		Ĭ
 포션	1	ed-shekel	Ī	stocked with the	ı	destroiss	I	적	1
HP : 1	6	Name of Street,	-	Autor a	uluis.	designation of	adre)e		

			하세요 동합니		p, dov	vn ,	,left	right,map,exit): right,
			적		USER	1	목적지	П
 아이템	I		I	I	 적	I		1
	I		I	I		Ī		1
	I	 적	 포션	I		Ī		1
 포션	I		I	I		I	 적	1
	를 <u>이</u>	로 이	하세요 동합니 적	다.		vn ,		,right,map,exit): right
 아이템	I		l	ı	 적	I		1
	ı		I	I		I		1
	ı	 적	포션	I		I		1
 포션	ı		I	I		I	 적	1
 HP: 1 목적지	4							

5. 결과 및 결론

- 1) 프로젝트 결과: up, down, left, right, map, exit 명령어를 받고, 지도를 출력하고, 목적지에 도착하는 등의 기능을 하는 MUD GAME을 완성함.
- 2) 느낀 점: 베이스 코드를 보고 먼저 주석의 중요성을 크게 깨달았음. 아무 것도 모른 상태로 코드를 보니 매우 막막한 데 주석을 통해 각 함수의 기능, 조건문이 들어간 이유 등을 쉽게 알 수 있었음.

베이스 코드를 기반으로 함수를 추가로 선언하거나, 변수를 추가하거나 등의 작업으로 프로그램을 직접 업그레이드해보니 구조를 파악하고 알고리즘을 짜는 과정에서 큰 흥미를 느낌. 곧 있을 개인 프로젝트도 이 MUD게임처럼 새로운 알고리즘을 짜서 간단하게 플레이 할 수 있는 게임을 주제로 프로젝트를 진행하고 싶음.