

1. 서론

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

2. 요구사항

- 사용자 요구사항 : 유저가 상하좌우로만 이동하며 목적지에 도착하는 게임
- 기능 계획
 1. user_HP변수를 전역변수로 만들어서 HP기능 추가(움직일 때 마다 HP감소)
 2. HP를 기본 20으로 설정한 후 0 이하가 될 시 게임 실패 종료.
 3. '아이템', '적', '포션'을 만나면 만났다는 상태를 출력.
적을 만나면 2 감소, 포션을 만나면 2 증가.
- 함수 계획
 1. printWrongMsg()함수를 만들어서 맵을 벗어날 경우 출력하는 함수.
 2. checkState()함수를 통해 유저의 상태를 파악하고 출력하는 함수.
 3. checkEnemy(), printEnemy() 적을 만났을 경우 체크 및 출력 함수.
 4. checkItem(), printCheck() 아이템을 만났을 경우 체크 및 출력 함수.
 5. checkPoshion(), printPoshion() 포션을 만났을 경우 체크 및 출력 함수

3. 설계 및 구현

- 기능 별 구현 사항

1. user_HP변수를 전역변수로 만들어서 HP기능 추가

```
const int mapX = 10;
const int mapY = 10;
int user_HP = 20;

if (user_input == "상") {
    // 위로 한 칸 올라가기
    user_y -= 1;
    bool inMap = checkXY(user_x, mapX, user_y, mapY);
    if (inMap == false) {
        printWrongMsg();
        user_y += 1;
    }
    else {
        cout << "위로 한 칸 올라갑니다." << endl;
        displayMap(map, user_x, user_y);
        user_HP--; // 이동할 때 HP 1씩 감소
    }
}
```

2. HP를 기본 20으로 설정한 후 0 이하가 될 시 게임 실패 종료.

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

3. '아이템', '적', '포션'을 만나면 만났다는 상태를 출력.
적을 만나면 2 감소, 포션을 만나면 2 증가.

```
// 적을 만났을 경우 출력하는 함수
void printEnemy(int map[][mapX], int user_x, int user_y) {
    cout << "적을 만났습니다. HP 2 줄어듭니다. " << endl;
}

// 아이템을 만났을 경우 출력하는 함수
void printItem(int map[][mapX], int user_x, int user_y) {
    cout << "아이템을 만났습니다." << endl;
}

// 포션을 만났을 경우 출력하는 함수
void printPotion(int map[][mapX], int user_x, int user_y) {
    cout << "포션을 만났습니다. HP 2 늘어났습니다. " << endl;
}
```

상태를 확인하는 함수

```
// 상태 확인하는 함수
void checkState(int map[][mapX], int user_x, int user_y) {
    // 적을 만났는지 확인
    if (checkEnemy(map, user_x, user_y)) {
        user_HP -= 2;
        printEnemy(map, user_x, user_y);
    }

    // 아이템을 만났는지 확인
    if (checkItem(map, user_x, user_y)) {
        printItem(map, user_x, user_y);
    }

    // 포션을 만났는지 확인
    if (checkPotion(map, user_x, user_y)) {
        user_HP += 2;
        printPotion(map, user_x, user_y);
    }
}
```

```

// 적을 만났는지 확인하는 함수
bool checkEnemy(int map[][mapX], int user_x, int user_y) {
    if (user_x == 2 && user_y == 0) {
        return true;
    }

    if (user_x == 0 && user_y == 1) {
        return true;
    }

    if (user_x == 3 && user_y == 1) {
        return true;
    }

    if (user_x == 4 && user_y == 4) {
        return true;
    }

    return false;
}

// 아이템을 만났는지 확인하는 함수
bool checkItem(int map[][mapX], int user_x, int user_y) {
    if (user_x == 1 && user_y == 0) {
        return true;
    }

    if (user_x == 0 && user_y == 1) {
        return true;
    }

    return false;
}

// 포션을 만났는지 확인하는 함수
bool checkPotion(int map[][mapX], int user_x, int user_y) {
    if (user_x == 0 && user_y == 4) {
        return true;
    }

    if (user_x == 2 && user_y == 3) {
        return true;
    }

    return false;
}

```

4. 테스트

- 기능별 테스트 결과 :

1. USER가 움직였을 경우 HP -1을 하고, 현재 HP 출력. 또한 아이템을 만났을 경우 출력.

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

	USER	적		목적지
아이템			적	
	적	포션		
포션				적

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

2. 적을 만났을 경우 HP -2, 적을 만났다고 출력.

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

	아이템	USER		목적지
아이템			적	
	적	포션		
포션				적

적을 만났습니다. HP 2 줄어듭니다.

3. 목적지에 도착했을 경우

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

아이템		적	USER	
아이템			적	
	적	포션		
포션			적	

목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.

4. HP가 0이 됐는데, 목적지에 도착 못했을 경우

현재 HP : 3 명령어를 입력하세요 (상,하,좌,우,지도,종료): 우
오른쪽으로 이동합니다.

아이템		USER	목적지	
아이템			적	
	적	포션		
포션			적	

적을 만났습니다. HP 2 줄어듭니다.
HP가 0 이하가 되었습니다. 실패했습니다.
게임을 종료합니다.

5. 결과 및 결론

1. 프로젝트 결과 : mud game을 만들었다.
2. 느낀 점: 이번 코드는 해결하는 과정에서 실수가 좀 발생해서 재밌었습니다.