

mud game 보고서

234106 임연우

1. 서론

1) 프로젝트 목적 및 배경

: 7 주차까지 배운 내용에 대한 적용과 응용을 위한 실습

2) 목표

: 지금까지 배운 내용을 다양하게 활용한 간단한 MUD 게임의 구현

2. 요구사항

1) 사용자 요구사항

: 유저가 상, 하, 좌, 우, 지도를 이용하여 목적지에 도착

2) 기능 계획

(1) 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

(2) 지도 밖으로 나가게 되면 에러 메시지 출력

(3) 목적지에 도착하면 “성공”을 출력하고 종료

-추가 기능 요구사항

(1) 유저는 체력 20을 가지고 게임 시작

(2) 사용자가 이동할 때 마다 사용자 체력 1씩 감소

(3) 처음 명령문을 입력 받을 때 마다 HP 함께 출력

(4) HP가 0이 되면 “실패”를 출력하고 종료

(5) 무기/갑옷, 포션, 적을 만났을 때 그에 대한 메시지를 출력

3) 함수 계획

(1) 메인 함수: 사용자에게 값을 계속 입력받고, 그에 대한 함수 호출

(2) 지도와 현재 위치 출력 함수: displayMap()

(3) 사용자 위치 체크 함수: checkXY()

(4) 목적지에 도착 체크 함수: checkGoal()

(5) 캐릭터의 위치를 옮기는 함수: userMoving()

(6) 캐릭터 위치에 있는 사물과 상호작용하는 함수: checkMap()

3. 설계 및 구현

요구사항1. 사용자에게 “상”, “하”, “좌”, “우”, “지도”, “종료” 중 하나를 입력 받기

좌표를 저장하는 클래스 locateXY 설정

```
struct locateXY {  
    int x;  
    int y;  
};
```

게임의 지도 설정

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} };
```

유저의 위치를 저장할 변수를 locateXY 클래스를 이용해서 초기화

```
locateXY user = {0, 0};
```

```
while (true) { // 사용자에게 계속 입력받기 위해 무한 루프  
    cout << "현재 체력은 " << userLP << "입니다" << endl;  
  
    // 사용자의 입력을 저장할 변수  
    string user_input = "";  
  
    cout << "명령어를 입력하세요 (up,down,left,right,map,quit): ";  
    cin >> user_input;  
  
    if (user_input == "map") {  
        displayMap(map, user.x, user.y);  
        userLP += 1;  
    }  
    else if (user_input == "quit") {  
        cout << "종료합니다.";  
        break;  
    }  
    else if (user_input == "up" || user_input == "down"  
    || user_input == "left" || user_input == "right") {  
        locateXY newLocation = userMoving(user.x, user.y, user_input);  
  
        if (checkXY(newLocation.x, newLocation.y)) {  
            user = newLocation;  
            displayMap(map, user.x, user.y);  
        }  
        else {  
            cout << "맵을 벗어났습니다. 다시 돌아갑니다" << endl;  
        }  
    }  
    else {  
        cout << "잘못된 입력입니다." << endl;  
        userLP += 1;  
    }  
}
```

1) 입력

- userLP : 유저의 체력
- user_input : 유저가 입력한 값
- map : 게임에 설정되어있는 맵
- user.x, user.y : 클래스 locateXY로 생성한 객체 user의 x, y 좌표 데이터
- newLocation.x, newLocation.y : 클래스 locateXY 형식의 이동한 후의 위치 좌표 데이터

2) 결과

- 현재 체력 출력
- 명령어 안내와 독촉
- 유저의 입력을 확인 후 입력에 맞는 설명 출력과 함수 호출

3) 설명

- while로 계속해서 사용자의 입력을 받음
- user_input을 입력받아 저장하고 if로 어떤 입력인지 확인
- if문으로, user_input이 "map" 이면, displayMap() 함수 호출해서 지도를 표시.
- while 한바퀴 돌 때마다 체력이 1 닳도록 해놨기 때문에, 지도 표시의 체력 소모를 없애기 위해 userLP를 +1.
- if문으로, user_input이 "quit"이면 종료를 안내하고 break로 게임 종료
- if문으로, user_input이 상, 하, 좌, 우 중 하나이면, userMoving() 함수를 호출해 이동한 좌표를 저장.
- if문으로 함수 checkXY()로 이동한 좌표가 유효한지 확인하고, 유효하다면 user에 저장하고, displayMap() 함수로 현재 지도 출력
- 유효하지 않다면, user에 저장하지 않고, 유효하지 않음을 안내.
- 입력한 내용이 어느것도 해당하지 않다면, 잘못된 입력임을 안내, 잘못된 입력의 체력 감소를 없애기 위해 체력 + 1.

```

locateXY userMoving(int user_x, int user_y, string user_input) {
    locateXY newLocation = {user_x, user_y};

    if (user_input == "up") {
        newLocation.y -= 1;
    } else if (user_input == "down") {
        newLocation.y += 1;
    } else if (user_input == "left") {
        newLocation.x -= 1;
    } else if (user_input == "right") {
        newLocation.x += 1;
    }

    return newLocation;
}

```

1) 입력

- user_x, user_y : 객체 user의 x, y 값을 받아오기 위한 매개변수
- user_input : 사용자의 입력
- newLocation : 클래스 locatXY로 생성한, 유저의 좌표를 저장하기 위한 변수

2) 결과

- 사용자 입력에 따른 이동한 좌표를 반환

3) 설명

- user_input이 "up"인지 if문으로 체크, 맞으면 y값 -1
- user_input이 "down"인지 if문으로 체크, 맞으면 y값 +1
- user_input이 "left"인지 if문으로 체크, 맞으면 x값 -1
- user_input이 "right"인지 if문으로 체크, 맞으면 x값 +1
- 이동한 좌표 반환

```

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 |"; // 양 옆 1칸 공백
            }
            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 << " ----- " << endl;
    }
}

```

1) 입력

- mapX, mapY : 전역변수로 설정된 맵의 가로, 세로 크기
- map[][mapX] : map 배열을 가져오기 위한 매개변수
- user_x, user_y : 객체 user 의 x, y 값을 받아오기 위한 매개변수
- posState: 맵에 위한 사물을 가져오는 변수

2) 결과

- 맵의 가로, 세로 만큼 반복해서 맵의 구성요소 출력

3) 설명

- for문으로 맵의 크기만큼 반복
- 맵의 요소를 posState에 저장해서 switch로 체크
- posState 가 0이면 빈칸, 1이면, 아이템, 2이면 아이덴, 3이면 포션, 4면 목적지 출력.

```

bool checkXY(int user_x, int user_y) {
    return (user_x >= 0 && user_x < mapX && user_y >= 0 && user_y < mapY)
}

```

1) 입력

- user_x, user_y : user의 x, y 값
- mapX, mapY : 맵의 가로, 세로 크기

2) 결과

- 비교식에 의한 bool값 반환

3) 설명

- if문으로, 이동한 값이 맵을 벗어나지 않았는지 확인하고, 불 값 반환

요구사항2. 지도 밖으로 나가게 되면 에러 메시지 출력

```
else if (user_input == "up" || user_input == "down"
|| user_input == "left" || user_input == "right") {
    locateXY newLocation = userMoving(user.x, user.y, user_input);

    if (checkXY(newLocation.x, newLocation.y)) {
        user = newLocation;
        displayMap(map, user.x, user.y);
    }
    else {
        cout << "맵을 벗어났습니다. 다시 돌아갑니다" << endl;
    }
}
```

- 요구사항1과 중첩
- 최적화하는 과정에서 요구사항 2를 요구사항 1에 포함하는 것이 낫겠다고 판단.
- 이 코드 구현에 대한 설명은 이미 요구사항 1에 쓰여있음

요구사항 3. 목적지에 도착하면 “성공”을 출력하고 종료

```
if( checkGoal(map, user.x, user.y)) {  
    cout << "목적지에 도착했습니다! 축하합니다!" << endl;  
    cout << "게임을 종료합니다." << endl;  
    break;  
}
```

1) 입력

- map : 게임에 설정되어있는 맵
- user.x, user.y : 클래스 locateXY로 생성한 객체 user의 x, y 좌표 데이터

2) 결과

- 목적지에 도착했음을 알리고
- 게임 종료

3) 설명

- checkGoal() 함수를 호출하여 if문으로 확인
- 목적지 도착이 맞으면 도착과, 게임 종료를 안내하고 종료.

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

1) 입력

- map[][mapX] : map 배열
- user_x, user_y : user의 x, y 값

2) 결과

- 지도의 유저 위치의 값에 따른 bool 값 반환

3) 설명

- 지도의 유저 위치의 값이 4이면 true, 4가 아니면 false 반환

추가 요구사항 1. 유저는 체력 20을 가지고 게임 시작

```
int userLP = 20;
```

- 유저의 체력을 userLP에 20으로 초기화

추가 요구사항 2. 사용자가 이동할 때마다 사용자 체력 1씩 감소

```
userLP -= 1;
```

- 반복문이 반복될 때마다, 1씩 감소

추가 요구사항 3. 처음 명령문을 입력받을 때마다 HP 함께 출력

```
while (true) { // 사용자에게 계속 입력받기 위해 무한 루프  
    cout << "현재 체력은 " << userLP << "입니다" << endl;
```

1) 입력

- userLP : 유저의 체력

2) 결과

-반복문 시작할 때마다, 현재 체력 안내.

추가 요구사항 4. HP가 0이 되면 “실패”를 출력하고 종료

```
if (userLP == 0) {  
    cout << "실패" << endl << "게임 종료";  
    break;  
}
```

1) 입력

- userLP : 유저의 체력

2) 결과

- 실패를 출력하고 게임종료

3) 설명

- if문으로 체력이 0인지 확인

추가 요구사항 5. 무기/갑옷, 포션, 적을 만났을 때 그에 대한 메시지를 출력

```
userLP = checkMap(map, user.x, user.y, userLP);
```

1) 입력

- map : 게임에 설정되어있는 맵

- user.x, user.y : 클래스 locateXY로 생성한 객체 user의 x, y 좌표 데이터

- userLP : 유저의 체력

2) 결과

- checkMap() 함수를 호출하고 결과를 userLP에 저장

3) 설명

- checkMap() 함수를 호출하고 결과를 userLP에 저장

```
int checkMap(int map[][mapX], int user_x, int user_y, int userLP) {
    int mapState = map[user_y][user_x];

    // 0은 빈 공간, 1은 아이템, 2는 적, 3은 포션, 4는 목적지
    switch (mapState) {
        case 1:
            cout << "아이템을 주웠습니다" << endl;
            break;
        case 2:
            cout << "적을 만났습니다" << endl << "LP 2 감소" << endl;
            userLP -= 2;
            break;
        case 3:
            cout << "포션을 마셨습니다" << endl << "LP 2 증가" << endl;
            userLP += 2;
            break;
    }
    return userLP;
}
```

1) 입력

- map[][mapX] : map 배열
- user_x, user_y : user의 x, y 값
- userLP : 유저의 체력
- mapState : 지도의 유저 위치의 값

2) 결과

- mapState 값에 따라 스위치의 값을 출력하고, 체력 변화 반환

3) 설명

- switch로 지도의 값에 따른 상호작용
- 1 이면 아이템을 출력
- 2이면 적을 출력하고, 체력 2 감소
- 3이면 포션을 출력하고, 체력 2 증가
- 변화한 체력을 반환

4. 테스트

1) 기능 별 테스트 결과

(1) 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

1. 맵

```
현재 체력은 16입니다
명령어를 입력하세요 (up,down,left,right,map,quit): map
|아이템|  적  |      |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      | USER |      |      |
-----
      |  적  | 포션  |      |      |
-----
포션  |      |      |      |  적  |
-----
```

2. 상

```
현재 체력은 16입니다
명령어를 입력하세요 (up,down,left,right,map,quit): up
|아이템|  적  |      |목적지|
-----
아이템|      | USER |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션  |      |      |
-----
포션  |      |      |      |  적  |
-----
```

3. 하

```
현재 체력은 15입니다
명령어를 입력하세요 (up,down,left,right,map,quit): down
|아이템|  적  |      |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      | USER |      |      |
-----
      |  적  | 포션  |      |      |
-----
포션  |      |      |      |  적  |
-----
```

4. 좌

```

현재 체력은 14입니다
명령어를 입력하세요 (up,down,left,right,map,quit): left
|아이템| 적 | |목적지|
-----
아이템|   |   | 적 |   |
-----
| USER |   |   |   |
-----
|  적  | 포션 |   |   |
-----
포션 |   |   |   |  적 |
-----

```

5. 우

```

현재 체력은 13입니다
명령어를 입력하세요 (up,down,left,right,map,quit): right
|아이템| 적 | |목적지|
-----
아이템|   |   | 적 |   |
-----
|   | USER |   |   |
-----
|  적  | 포션 |   |   |
-----
포션 |   |   |   |  적 |
-----

```

6. 잘못된 입력

```

현재 체력은 12입니다
명령어를 입력하세요 (up,down,left,right,map,quit): qwer
잘못된 입력입니다.
현재 체력은 12입니다
명령어를 입력하세요 (up,down,left,right,map,quit): █

```

(2) 지도 밖으로 나가게 되면 에러 메시지 출력

```

명령어를 입력하세요 (up,down,left,right,map,quit): right
|아이템| 적 | |목적지|
-----
아이템|   |   | 적 |   |
-----
|   |   |   | USER |
-----
|  적  | 포션 |   |   |
-----
포션 |   |   |   |  적 |
-----
현재 체력은 10입니다
명령어를 입력하세요 (up,down,left,right,map,quit): right
맵을 벗어났습니다. 다시 돌아갑니다
현재 체력은 9입니다
명령어를 입력하세요 (up,down,left,right,map,quit): █

```

(3) 목적지에 도착하면 “성공”을 출력하고 종료

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

```
현재 체력은 8입니다
명령어를 입력하세요 (up,down,left,right,map,quit): up
|아이템|  적  |      | USER |
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.
```

-추가 기능 요구사항

(1) 유저는 체력 20을 가지고 게임 시작

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\CPP2409> & 'c:\Users\user2\.vscode\extensions\ms-vscode.cpptools-1.22.1
0-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MI
Engine-In-ng3khgh1.uec' '--stdout=Microsoft-MIEngine-Out-qdaqmzx1.r1x' '--stder
r=Microsoft-MIEngine-Error-kpp0nxnj.klv' '--pid=Microsoft-MIEngine-Pid-c510c3g
4.fnw' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
현재 체력은 20입니다
명령어를 입력하세요 (up,down,left,right,map,quit):
```

(2) 사용자가 이동할 때 마다 사용자 체력 1씩 감소

```

현재 체력은 20입니다
명령어를 입력하세요 (up,down,left,right,map,quit): down
  |아이템|  적  |      |목적지|
-----
USER |      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
아이템을 주웠습니다
현재 체력은 19입니다
명령어를 입력하세요 (up,down,left,right,map,quit): down
  |아이템|  적  |      |목적지|
-----
아이템|      |      |  적  |      |
-----
USER |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
현재 체력은 18입니다
명령어를 입력하세요 (up,down,left,right,map,quit): █
  
```

(3) 처음 명령문을 입력 받을 때 마다 HP 함께 출력

```

현재 체력은 20입니다
명령어를 입력하세요 (up,down,left,right,map,quit): map
USER |아이템|  적  |      |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
현재 체력은 20입니다
명령어를 입력하세요 (up,down,left,right,map,quit): right
  | USER |  적  |      |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
아이템을 주웠습니다
현재 체력은 19입니다
명령어를 입력하세요 (up,down,left,right,map,quit): █
  
```

(4) HP가 0이 되면 “실패”를 출력하고 종료

```

현재 체력은 4입니다
명령어를 입력하세요 (up,down,left,right,map,quit): right
|아이템| 적 | USER |목적지|
-----
아이템|   |   |   |   |
-----
|   |   |   |   |
-----
|   |   |   |   |
-----
포션 |   |   |   |   |
-----
현재 체력은 3입니다
명령어를 입력하세요 (up,down,left,right,map,quit): down
|아이템| 적 |   |목적지|
-----
아이템|   |   |   |   |
-----
|   |   |   |   |
-----
|   |   |   |   |
-----
포션 |   |   |   |   |
-----
적을 만났습니다
LP 2 감소
실패
게임 종료
PS C:\CPP2409>

```

(5) 무기/갑옷, 포션, 적을 만났을 때 그에 대한 메시지를 출력

1. 아이템

```

현재 체력은 18입니다
명령어를 입력하세요 (up,down,left,right,map,quit): right
|   | USER |   |   |목적지|
-----
아이템|   |   |   |   |
-----
|   |   |   |   |
-----
|   |   |   |   |
-----
포션 |   |   |   |   |
-----
아이템을 주웠습니다
현재 체력은 19입니다

```

2. 적

```

현재 체력은 17입니다
명령어를 입력하세요 (up,down,left,right,map,quit): right
|아이템| USER |   |목적지|
-----
아이템|   |   |   |   |
-----
|   |   |   |   |
-----
|   |   |   |   |
-----
포션 |   |   |   |   |
-----
적을 만났습니다
LP 2 감소
현재 체력은 14입니다
명령어를 입력하세요 (up,down,left,right,map,quit):

```


3, 포션

```
현재 체력은 12입니다
명령어를 입력하세요 (up,down,left,right,map,quit): down
|아이템|  적  |      |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | USER |      |      |
-----
포션 |      |      |      |  적  |
-----
포션을 마셨습니다
LP 2 증가
현재 체력은 13입니다
명령어를 입력하세요 (up,down,left,right,map,quit): █
```


2) 최종 테스트 결과

```
현재 체력은 20입니다
명령어를 입력하세요 (up,down,left,right,map,quit): map
  USER |아이템|  적  |      |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
현재 체력은 20입니다
명령어를 입력하세요 (up,down,left,right,map,quit): right
      | USER |  적  |      |목적지|
-----
아이템|      |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
아이템을 주웠습니다
```

```
아이템을 주웠습니다
현재 체력은 19입니다
명령어를 입력하세요 (up,down,left,right,map,quit): down
      |아이템|  적  |      |목적지|
-----
아이템| USER |      |  적  |      |
-----
      |      |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
현재 체력은 18입니다
명령어를 입력하세요 (up,down,left,right,map,quit): down
      |아이템|  적  |      |목적지|
-----
아이템|      |      |  적  |      |
-----
      | USER |      |      |      |
-----
      |  적  | 포션 |      |      |
-----
포션 |      |      |      |  적  |
-----
```

현재 체력은 17입니다
명령어를 입력하세요 (up,down,left,right,map,quit): down
|아이템| 적 | |목적지|

아이템| | | 적 | |

| | | | |

| USER | 포션 | | |

포션 | | | | 적 |

적을 만났습니다
LP 2 감소
현재 체력은 14입니다
명령어를 입력하세요 (up,down,left,right,map,quit): right
|아이템| 적 | |목적지|

아이템| | | 적 | |

| | | | |

| 적 | USER | | |

포션 | | | | 적 |

포션을 마셨습니다
LP 2 증가
현재 체력은 15입니다
명령어를 입력하세요 (up,down,left,right,map,quit): right

현재 체력은 15입니다
명령어를 입력하세요 (up,down,left,right,map,quit): right
|아이템| 적 | |목적지|

아이템| | | 적 | |

| | | | |

| 적 | 포션 | USER | |

포션 | | | | 적 |

현재 체력은 14입니다
명령어를 입력하세요 (up,down,left,right,map,quit): right
|아이템| 적 | |목적지|

아이템| | | 적 | |

| | | | |

| 적 | 포션 | | USER |

포션 | | | | 적 |

```

현재 체력은 13입니다
명령어를 입력하세요 (up,down,left,right,map,quit): up
|아이템| 적 | |목적지|
-----
아이템| | | 적 | |
-----
| | | | USER |
-----
| 적 | 포션 | | |
-----
포션 | | | | 적 |
-----
현재 체력은 12입니다
명령어를 입력하세요 (up,down,left,right,map,quit): up
|아이템| 적 | |목적지|
-----
아이템| | | 적 | USER |
-----
| | | | |
-----
| 적 | 포션 | | |
-----
포션 | | | | 적 |
-----
현재 체력은 11입니다
명령어를 입력하세요 (up,down,left,right,map,quit): qwer
잘못된 입력입니다.

```

```

현재 체력은 11입니다
명령어를 입력하세요 (up,down,left,right,map,quit): up
|아이템| 적 | | USER |
-----
아이템| | | 적 | |
-----
| | | | |
-----
| 적 | 포션 | | |
-----
포션 | | | | 적 |
-----
목적지에 도착했습니다! 축하합니다!
게임을 종료합니다.
PS C:\CPP2409> 

```

5. 결과 및 결론

1) 프로젝트 결과

: 7주차 까지 학습한 내용을 바탕으로 Mud game을 만들었다.

if문, 반복문, switch, 클래스, 함수화 등 다양한 개념을 활용하면서 깊은 이해도를 가지게 되었다.

2) 느낀점

: 이번 프로젝트는 과제라는 느낌보다는 놀이라는 느낌이 더 들었다. 아주 간단한 게임이지만, 게임을 구현한다는 그 사실이 나를 들뜨게 한 것 같다. \

프로젝트를 진행하면서 다양한 아이디어가 떠올라서 하나씩 적용해보기도 했는데, 주어진 요구사항만을 제대로 하는게 프로젝트로서 옳은 것 같아, 이 프로젝트에서는 아이디어를 배제해서 진행했다.

대신 새로운 파일을 만들어서 그 파일에는 여러 가지 시도를 해보는 것도 재밌을 것 같다.