

# NEFUS GAME PROJECT

---

10115 박인성



# 동기

---

후...

네퓨즈 2차 설명회에서 본 미로 찾기가 생각이 나서  
이를 계기로 미로 찾기를 하기로 했습니다



출처 : 네퓨즈 공식 페이스북 페이지

구현 방법

## Easy (쉬움)

## Normal (보통)

## Hard (어려움)

## 총 3개의 단계를 배열로 나누어서 맵을 구성

## 배열에서 0, 1, 2, 3을 활용함

0 = \* (벽)

1 = (공백)

2 = \$ (도착 지점)

### 3 = @ (플레이어)

# 구현 방법

캐릭터의 이동

( printMove(); / 맵 출력 함수 설명 )

KeyValue 키 눌림에 대한 값을  
\_getch()로 입력 받음

keyValue 으로 받은 키 눌림에 값을 활용하여  
X 값을 증가 혹은 감소,  
Y 값을 증가 혹은 감소 시키며

@ (사용자)의 움직임 좌표 설정

```
int looping = 0;  
char keyValue = _getch();
```

```
switch (keyValue) {  
case 'w':  
    --move_y;  
    break;
```

```
case 'a':  
    --move_x;  
    break;
```

```
case 'd':  
    ++move_x;  
    break;
```

```
case 's':  
    ++move_y;  
    break;
```

```
case 'q':  
    exit(1);  
    break;
```

```
default:  
    break;
```

# 구현 방법

캐릭터의 이동

( printMove(); / 맵 출력 함수 설명 )

move\_y = 이동한 y에 대한 값  
move\_x = 이동한 x에 대한 값

```
int looping = 0;  
char keyValue = _getch();
```

```
switch (keyValue) {  
case 'w':  
    --move_y;  
    break;
```

```
case 'a':  
    --move_x;  
    break;
```

```
case 'd':  
    ++move_x;  
    break;
```

```
case 's':  
    ++move_y;  
    break;
```

```
case 'q':  
    exit(1);  
    break;
```

```
default:  
    break;
```

# 구현 방법

캐릭터의 이동

( printMove(); / 맵 출력 함수 설명 )



**A**

왼쪽으로 이동  
= Y 값 감소

**W**

위로 이동  
= X 값 감소

**D**

오른쪽으로 이동  
= Y 값 증가

**S**

아래로 이동  
= X 값 증가

```
int looping = 0;  
char keyValue = _getch();
```

```
switch (keyValue) {  
case 'w':  
    --move_x;  
    break;
```

```
case 'a':  
    --move_y;  
    break;
```

```
case 'd':  
    ++move_y;  
    break;
```

```
case 's':  
    ++move_x;  
    break;
```

```
case 'q':  
    exit(1);  
    break;
```

```
default:  
    break;
```



# 구현 방법

캐릭터의 이동

( printMove(); / 맵 출력 함수 설명 )

30 x 30 배열 된 맵을 루프로 세팅 및 출력

```
int random = 0;
for (int x = 0; x < 30; x++) {
    for (int y = 0; y < 30; y++) {
        random = hardMap[x][y];

        if (hardMap[move_x][move_y] == 0) {
            crash();
            exit(1);
        }

        if (hardMap[move_x][move_y] == 2) {
            ending();
        }

        if (x == move_x && y == move_y) {
            hardMap[x][y] = 3;
        }

        switch (random) {
        case 0:
            printf(" *");
            break;
        case 1:
            printf("  ");
            break;
        case 2:
            printf(" $");
            break;
        case 3:
            printf(" @");
            hardMap[move_x][move_y] = 1;
            break;
        }

        printf("\n");
    }
}
```



# 구현 방법

캐릭터의 이동

( printMove(); / 맵 출력 함수 설명 )

루프를 돌면서 random 이라는 변수 안에  
현재 Map의 숫자를 갖고옴

캐릭터가 움직인 좌표에 벽이 있는지 없는지  
충돌 체크를 함.

캐릭터가 움직인 좌표가 최종 지점인지  
아닌지 확인 함

```
int random = 0;
for (int x = 0; x < 30; x++) {
    for (int y = 0; y < 30; y++) {
        random = hardMap[x][y];

        if (hardMap[move_x][move_y] == 0) {
            crash();
            exit(1);
        }

        if (hardMap[move_x][move_y] == 2) {
            ending();
        }

        if (x == move_x && y == move_y) {
            hardMap[x][y] = 3;
        }

        switch (random) {
        case 0:
            printf(" *");
            break;
        case 1:
            printf(" ");
            break;
        case 2:
            printf(" $");
            break;
        case 3:
            printf(" @");
            hardMap[move_x][move_y] = 1;
            break;
        }

        printf("\n");
    }
}
```

# 구현 방법

캐릭터의 이동

( printMove(); / 맵 출력 함수 설명 )

캐릭터가 움직인 자리를 3 (공백)으로 바꿈

맵에서 0인 부분은 벽 (\*)으로 출력  
1인 부분은 공백 ( )으로 출력  
2인 부분은 최종 목표 지점 (\$)으로 출력

3인 부분을 캐릭터로 출력,  
캐릭터가 있던 자리를 1로 바꿈  
=> 기존에 있던 자리를 지움

```
int random = 0;
for (int x = 0; x < 30; x++) {
    for (int y = 0; y < 30; y++) {
        random = hardMap[x][y];

        if (hardMap[move_x][move_y] == 0) {
            crash();
            exit(1);
        }

        if (hardMap[move_x][move_y] == 2) {
            ending();
        }

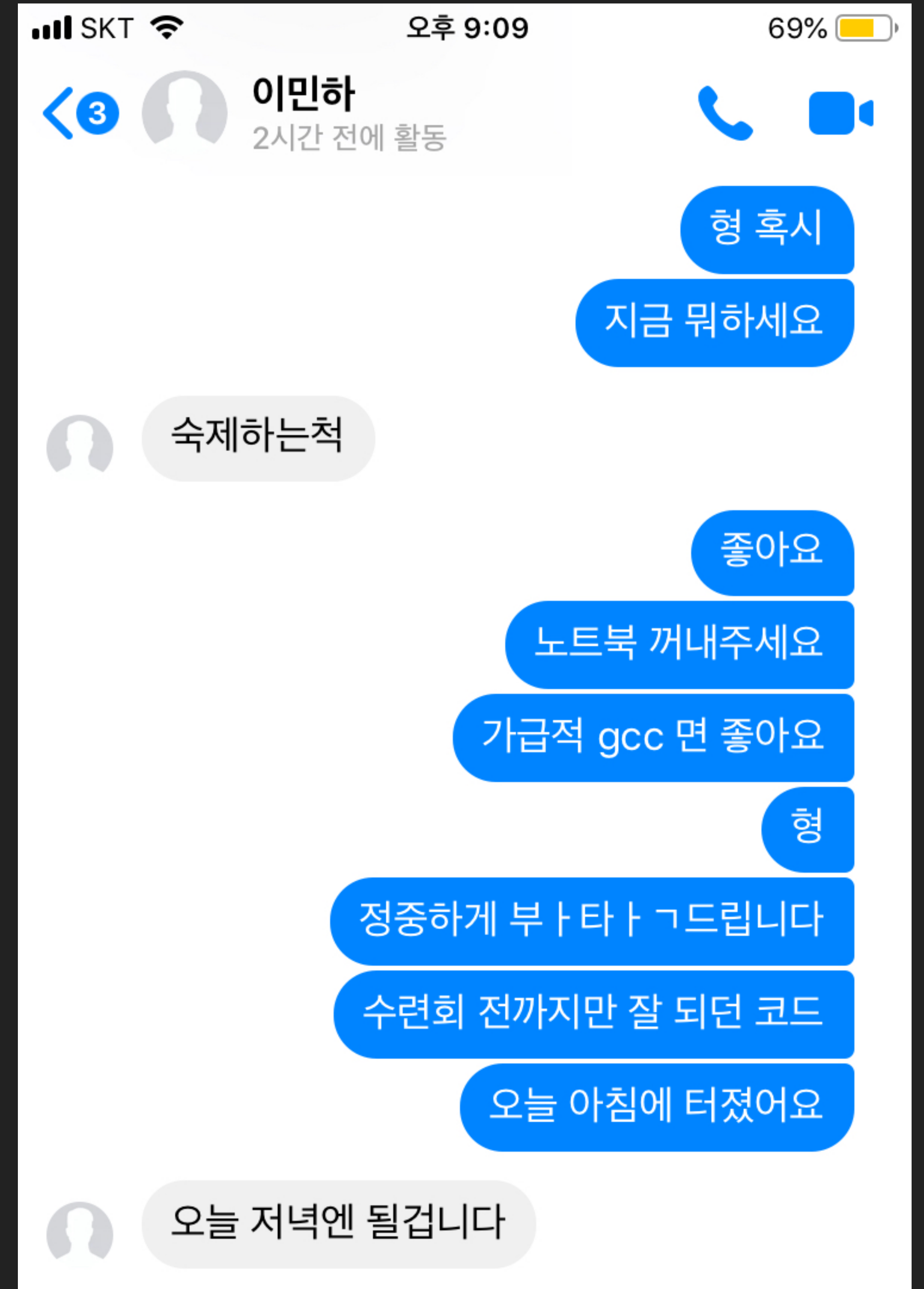
        if (x == move_x && y == move_y) {
            hardMap[x][y] = 3;
        }

        switch (random) {
            case 0:
                printf(" *");
                break;
            case 1:
                printf(" ");
                break;
            case 2:
                printf(" $");
                break;
            case 3:
                printf(" @");
                hardMap[move_x][move_y] = 1;
                break;
        }

        printf("\n");
    }
}
```

# 소감

“오늘 저녁에는 될겁니다”

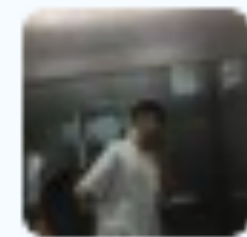


소감

“오늘 저녁에는 될겁니다”

# 진짜 아침에 못 찾은 에러

드디어 다 했습니다... ...



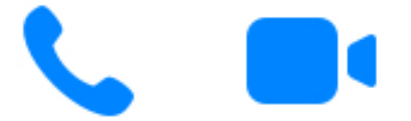
insung3511 committed a day ago

Some Update ...



insung3511 committed a day ago

SKT 9:09 69% 이민하



형 혹시

지금 뭐하세요

좋아요

노트북 꺼내주세요

급적 gcc 면 좋아요

형

부탁드립니다

하지만 잘 되던 코드

# 그 날 저녁에 에러 찾고 다 했습니다.

