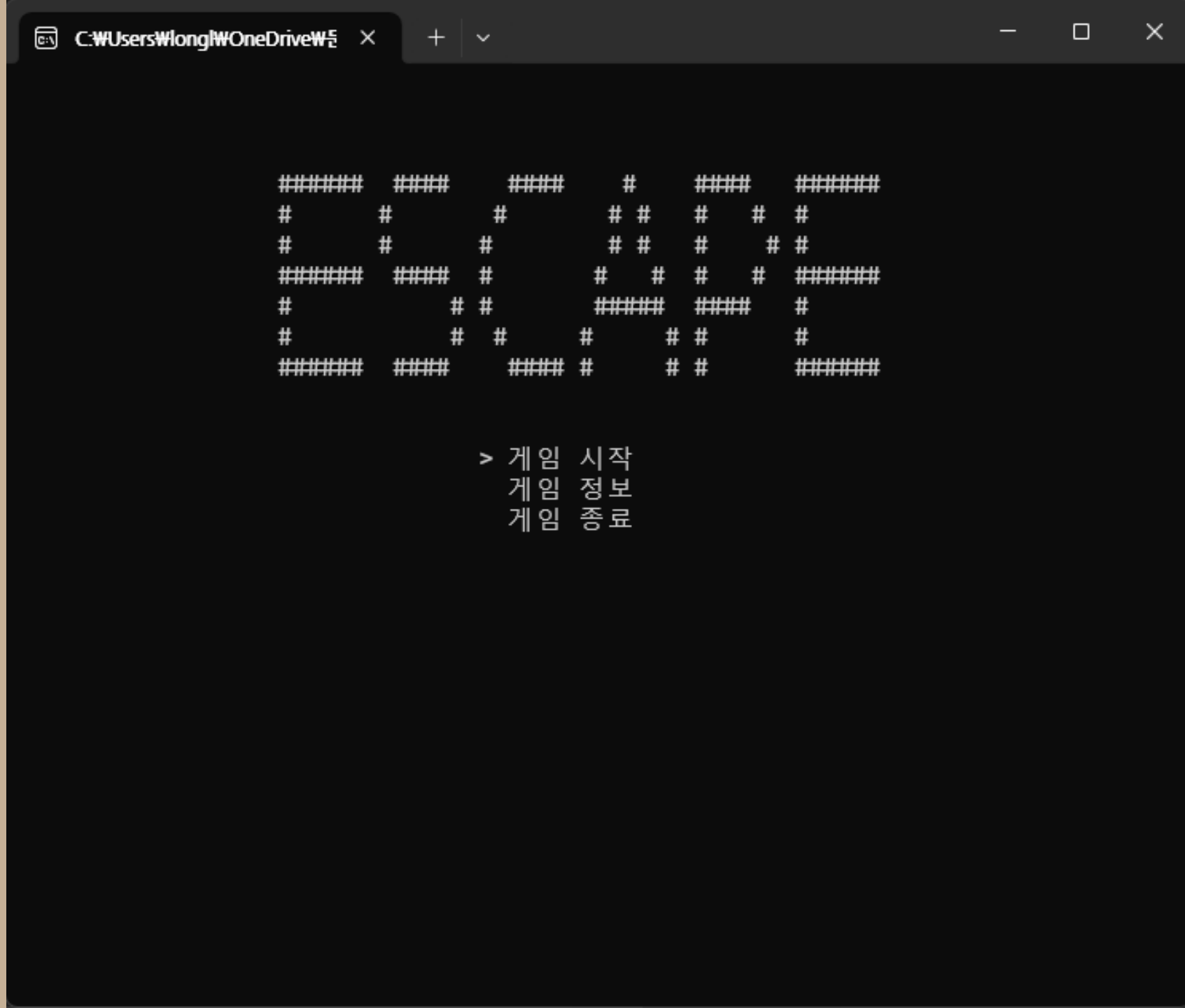
C++ 콘솔 게임 제작

장수덕

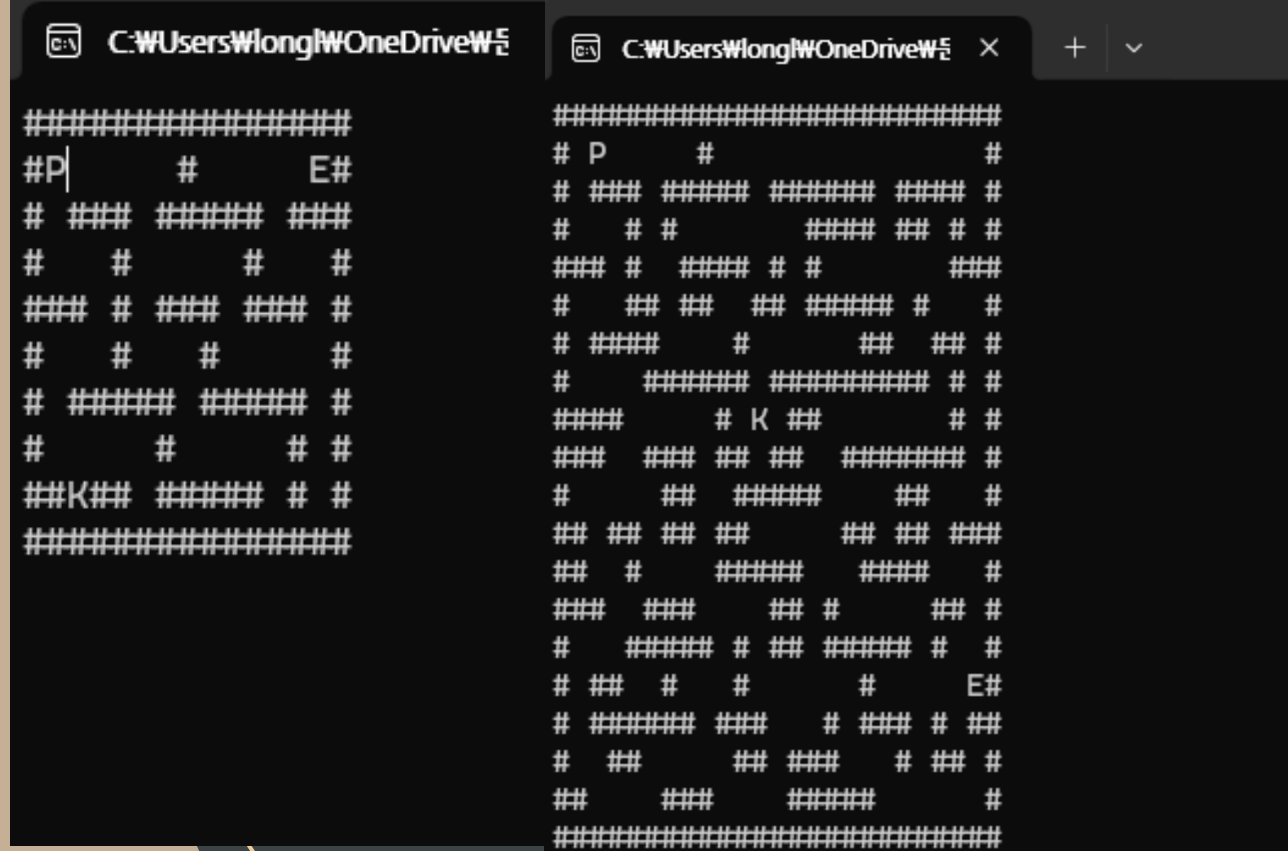
게임 소개

- 게임명 : ESCAPE
- 장르 : 미로 탈출
- 개발 환경 : c++콘솔



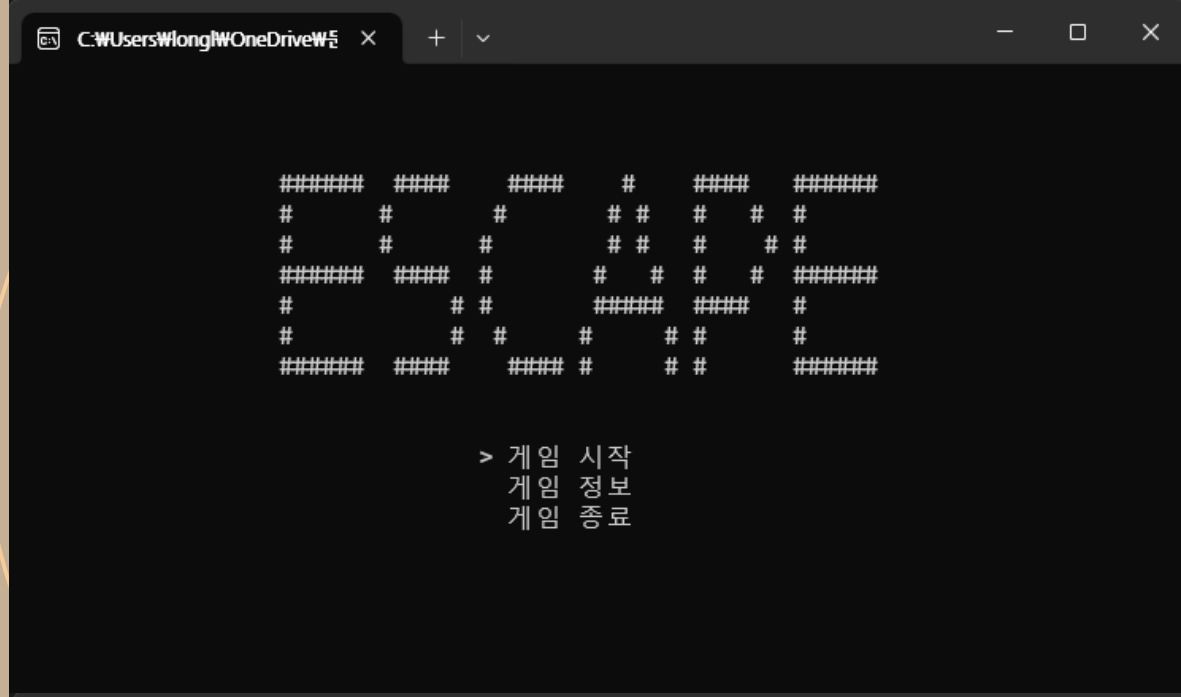
게임의 목표

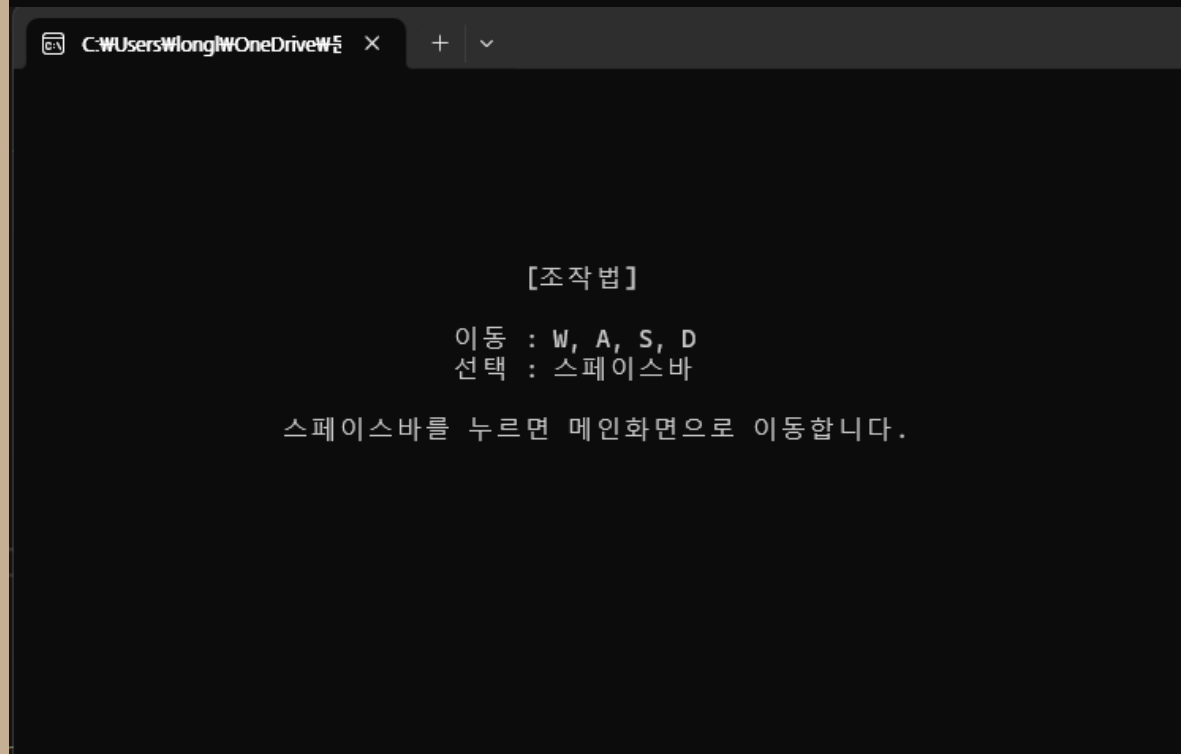
- 제한된 공간 안에서 출구를 찾아 탈출
- 간단한 조작으로 열쇠를 찾아 지정된 탈출구로 이동
- 난이도는 점진적으로 증가



게임 플레이 흐름

- 프로그램 실행 시 타이틀 제목과 메뉴가 생성
- 메뉴 구성 : 게임 시작, 게임 정보(조작법) 게임 종료
- 원하는 메뉴를 키보드 이동키로 이동한후 스페이스바를 눌러 실행





인터페이스 및 조작법

- 키보드 방향키로 이동가능
- 선택은 스페이스바
- 열쇠는 K 출구는 E 벽은 # 표시

주요 기술 구조

- 메뉴선택
- 이동 및 충돌
- 열쇠 획득 및 탈출

메뉴 선택 로직

- 함수 호출과 동시에 시작 좌표 값을 인수로 보냄
- 스페이스 바 입력시 현재 좌표값에서 12를 뺀 값을 리턴하여 변수에 저장

```
while (1)
{
    TitleDraw();
    int menucode = menu.draw(34, 12);

    if (menucode == 0)  // 게임 시작
    {
        break;
    }
    else if (menucode == 1)     // 게임 정보
    {
        menu.InfoDraw();
    }
    else if (menucode == 2)     // 게임 종료
    {
        return;
    }
}
```

```
int draw(int x, int y)
{
    gotoxy(x - 2, y);
    cout << "> 게임 시작";
    gotoxy(x, y + 1);
    cout << "게임 정보";
    gotoxy(x, y + 2);
    cout << "게임 종료";

    while (1)
    {
        int n = Keycontrol();

        switch (n)
        {
        case UP:
            if (y > 12)
            {
                gotoxy(x - 2, y);
                printf(" ");
                gotoxy(x - 2, --y);
                printf(">");
            }
            break;

        case DOWN:
            if (y < 14)
            {
                gotoxy(x - 2, y);
                printf(" ");
                gotoxy(x - 2, ++y);
                printf(">");
            }
            break;

        case SUBMIT: return y - 12;
```

이동 및 충돌

- 키보드 입력이 발생하면
- Move함수를 호출함과 동시에
- 인자를 함수쪽으로 보냄
- 현재 좌표값xy와 인수로 받은 이동값을 연산하여 다음 배열에 있는 문자를 switch로 판별
- 각 알맞은 충돌에 맞게 명령을 수행

```
while (running)
{
    int input = Keycontrol();

    switch (input)
    {
    case UP: win = player.move(maze, 0, -1, stage); break;
    case DOWN: win = player.move(maze, 0, 1, stage);  break;
    case LEFT: win = player.move(maze, -1, 0, stage); break;
    case RIGHT: win = player.move(maze, 1, 0, stage);  break;

    }
```

```
int move(Maze& maze, int dx, int dy, int s)
{

    int newX = x + dx;
    int newY = y + dy;
    char next;
```

```
switch (s)
{
case 1: next = maze.map1[newY][newX];

    if (next == '#')
    {
        return 0;
    }

    if (next == 'K')
    {
        gotoxy(x, y); cout << ' ';
        maze.map1[y][x] = ' ';
        x = newX;
        y = newY;
        maze.map1[y][x] = 'P';
        gotoxy(x, y); cout << 'P';

        return 1;
    }

    if (next == 'E')
    {
        return 2;
    }

    gotoxy(x, y); cout << ' ';
    maze.map1[y][x] = ' ';
    x = newX;
    y = newY;
    maze.map1[y][x] = 'P';
    gotoxy(x, y); cout << 'P';

    break;
```

KEY획득 및 탈출 로직

Move함수에서 다음 배열의 문자를 보고 판별한 리턴값을 변수에 받아 If문으로 열쇠를 획득 및 탈출을 판별

```
switch (input)
{
case UP: win = player.move(maze, 0, -1, stage); break;
case DOWN: win = player.move(maze, 0, 1, stage);  break;
case LEFT: win = player.move(maze, -1, 0, stage); break;
case RIGHT: win = player.move(maze, 1, 0, stage);  break;

}

if (win == 1)          // 열쇠 획득
{
    gotoxy(0, Maze :: rows + 2);
    cout << "열쇠 획득";
    clear++;

}

if (win == 2 && clear == 1)    // 탈출 로직
{
    system("cls");
    gotoxy(0,0);
    cout <<"다음 스테이지";
    stage++;
    Sleep(1000
    );
    break;
}
```

다음 스테이지

- 다음 스테이지로 이동하는 코드의 경우 함수 호출시 전달하는 인수의 값만 바꿔 다음 스테이지를 호출하는 방식

```
system("cls");
maze.draw(2);
player.draw(maze, 2);
player.Reset();


while (running)
{
    int input = Keycontrol();


    switch (input)
    {
    case UP: win = player.move(maze, 0, -1, stage); break;
    case DOWN: win = player.move(maze, 0, 1, stage);  break;
    case LEFT: win = player.move(maze, -1, 0, stage); break;
    case RIGHT: win = player.move(maze, 1, 0, stage);  break;
    }

    if (win == 1)
    {
        gotoxy(0, Maze::rows2 + 2);
        cout << "열쇠 획득";
        clear++;
    }

    if (win == 2 && clear == 2)
    {
        system("cls");
        gotoxy(0,0);
        cout <<"탈출 성공!";
        running = 0;
    }

}
```

실제 플레이

어려웠던 점

- 프로그래밍이 완전 처음이라 매우 미숙
- 처음 해본 콘솔 게임 개발
- 입출력 처리, 캐릭터 이동, 미로구현 등 모든 것이 도전이자 과제

해결 방법

- 유튜브/인터넷 학습
- AI 활용(ChatGPT 등)
- 자신이 이해할 수 있는 수준의 코드부터 시작
- 반복 연습 > 작은 성공 경험 > 점차 확장

마무리 및 느낀점

- 콘솔 게임이지만 스스로 만든 첫 결과물이라는 점에서 의미가 큼
- 처음에 전혀 갈피를 잡지 못한채 시간을 보내서 퀄리티적으로 아쉬움
- 타이머 기능, 자동으로 미로를 생성해주는 기능 등등 추가적으로 넣을 수 있는 기능을 추후에 넣어볼 예정

감사합니다