

Text Maze

H1105 김유민



1	프로젝트 구상 & 개발	10. 28 ~ 11. 17
2	디버깅 / 일부 기능 수정	11. 18 ~ 11. 22
3	문서화(발표자료 & pdf 등)	11. 18 ~ 11. 24

표 1 프로젝트 과정 (2018년도)

1. 개요

Text Maze를 만들며 고민했던 부분, 구현 과정, 디버그 작업 등을 정리하여 이곳에 적는다.

2. 목표

기존에 만들고 싶었던 게임 하나가 있었다. 마침 학교에서 프로젝트를 요구하여, 이번 기회에 제대로 포트폴리오를 만드는 기회로 삼고자 한다. C++을 이용하여 미로 게임을 만들어 봄으로써 ‘게임 개발’ 정신을 함양하고, 기존의 지식과 새로운 지식을 접목하여 기존의 포트폴리오를 뛰어넘는 작품을 만든다. 제대로 된 포트폴리오 하나가 부족한 시점에서 프로젝트형 수행평가는 둘도 없는 기회다. 특히 ‘기획’ 과정을 적음으로써 한 차원 높은 수준의 포트폴리오가 나올 것으로 예상된다. 제작 과정을 기록함으로써 다른 사람이 내가 만든 작품을 쉽게 이해하도록 서술하여 괜찮은 가치를 지니는 포트폴리오로 만들어 보고자 한다.

3. 프로그램의 시작

2학기 중간고사 이전부터 강의¹⁾를 보며 약간 만든 프로그램이 있었다. 만들고 싶었으나, 잠시 시험 때문에, 다른 공부로 미루던 것을 이번 수행평가를 계기로 일단 강의에서 하는 대로 그대로 따라서 만들었다. 전문가의 소스를 통해서 맵을 만드는 방법, 플레이어의 움직임 처리 등을 구현했고, 이것이 근간이 되어 점점 살을 붙여 나갔다.

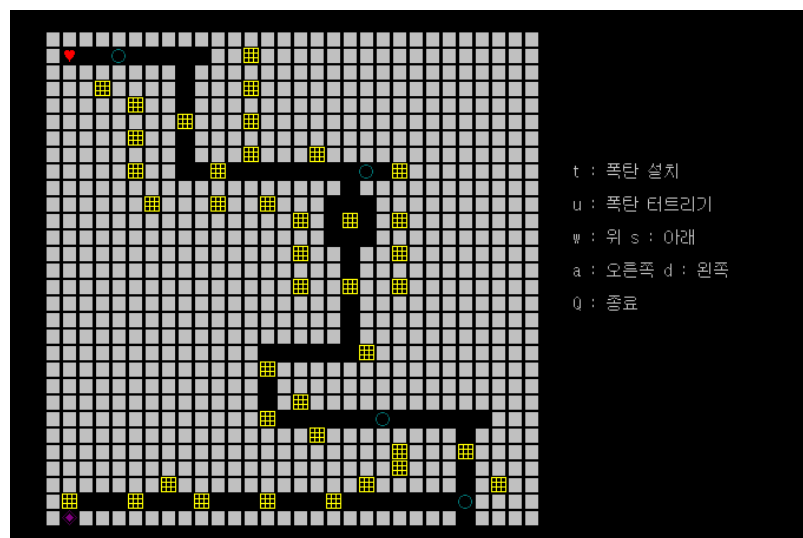


그림 2 초기의 모습 (스테이지 1) - 10.28

1) C언어/C++ 강의 38화 미로게임 1 미로설정과 출력 [어소트랙 게임아카데미]

무언가를 기획하고 코드를 쓰기 시작한 것이 아니라, 일단 짜면서 어느 방향으로 내가 원하는 것을 구현할지를 고민했다. 기본적인 움직임 다음에²⁾ 추가한 것이 화살처리였다. 한번 유튜브를 통해서 텍스트 기반의 화살 피하기 게임을 봤었다. 그 기억을 토대로 그곳에서 있던 두 기능³⁾을 꼭

```
cout << i << " ( " << arrowPos[i].x << ", "
    << arrowPos[i].y << " )" << arrowVector[i] << endl;
_getch();
arrowPos[i].x++;
if (Maze[arrowPos[i].y][arrowPos[i].x] == 'e')//이동해보니
    arrowPos[i] = arrowStart[i];
Maze[arrowPos[i].y][arrowPos[i].x] = '>';
cout << i << " ( " << arrowPos[i].x << ", "
    << arrowPos[i].y << " )" << arrowVector[i] << endl;
_getch();
```

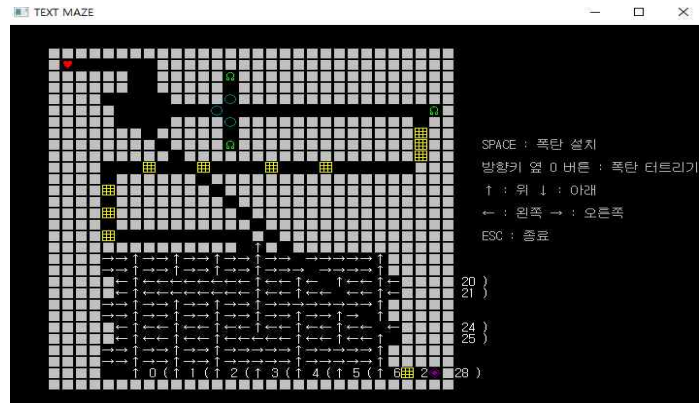
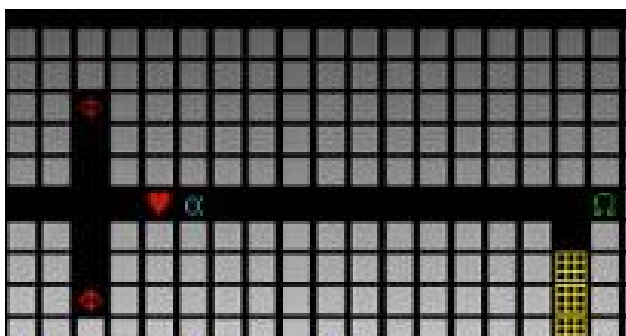


그림 4 arrow error (스테이지 2) - 10.31

구현하고 싶었다.

화살에 대한 처리를 해주면서 처음에는 화살의 위치를 없애는 처리를 하지 않고 찍어주는 바람에 화살의 수가 계속 증가하는 오류가 생겼다. 이것을 고치자 이번에는 화살이 (그래도) 많아서 느린 탓에 스레드의 개념을 찾던 때가 있었다.⁴⁾

스레드의 불필요를 느끼고⁵⁾ 더는 스레드를 찾지 않았다. 스레드를 통해 1과 2가 불규칙하게 수행되는 프로그램을 만들었는데, 문제는 이걸 적용하려 하니 하는 방법을 모르고, 괜히 꼬인다는 생각에 그만두었다. 그 이후로 스위치가 있으면 좋을 것 같다는 생각이 들어서 공을 굴려 스위치를 키는 방법을 생각하여 적용했다. 스위치를 가지고 도착점을 열어줄 터라 개수를 세는 처리를 했으며, 화살에 맞으면 시작점으로 이동하는 '죽음' 처리를 만들었다.



과자 블록은 폭탄으로 없앨 수 있는 물체
 α 는 공, ϕ 는 스위치가 켜진 상태

지금과는 많이 다른 초기의 설정 - 11.3

-
- 2) 폭탄/움직임은 그대로 강의를 따라서 만들었다. 하지만, 추구하는 방향성의 차이로 나는 내 생각을 붙이기 시작했다. (강의에서는 크레이지 아케이드를 만든다.)
 - 3) 화살/굴리는 블록(이것을 통해서 화살을 막는 그런 용도였다.)
 - 4) 모든 화살을 if문을 가지고 처리를 하니 하나만 증가하여도 속도가 느렸다.
 - 5) `_kbhit()`이 성립하지 않으면 화살에 대한 처리만 해주니 속도가 빨라졌다.

4. 프로그램의 변천

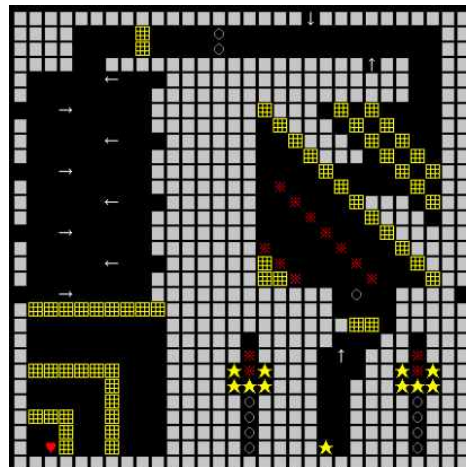
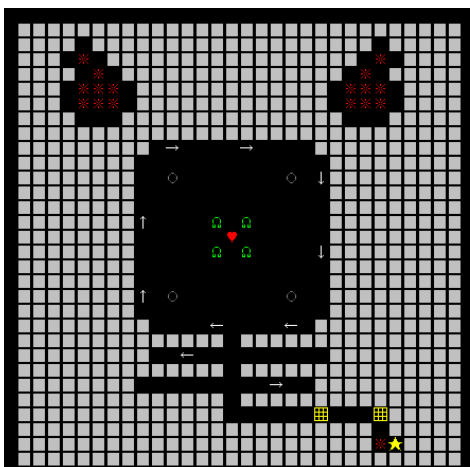
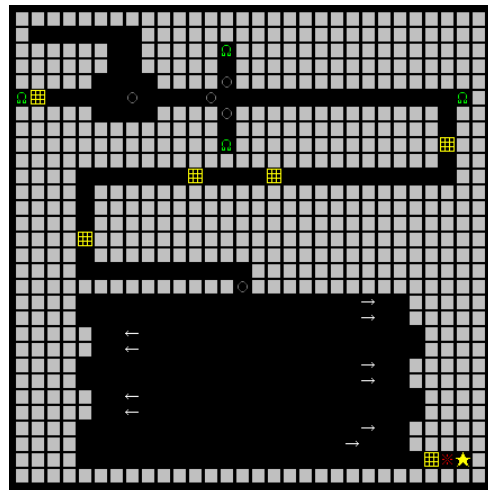
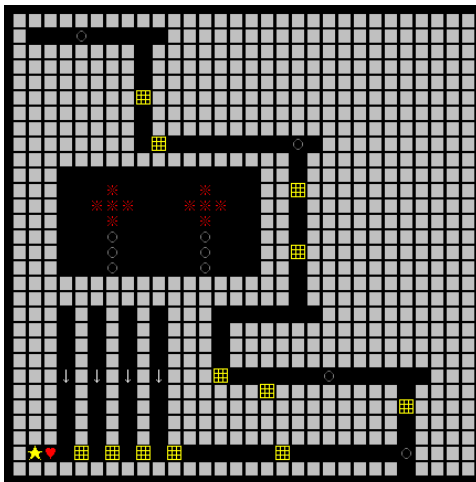
1. 데이터 처리(함수 소스)와 main을 완전히 분리하여 속도 저하 최소화.
2. BackSpace ← 를 통한 재시작
3. 스위치와 연동되는 블럭 9번(EndPos - blocking block) 오류 해결 (setmaze 후에 바로 output)
4. 화살과 폭탄이 만남. -> FIRE
5. 특수문자 change 색채 change
6. arrow 앞에 push블럭이 있는 경우 처리. 못가도록.



11월 5일 자 업데이트 내역과 6번에 대한 사진.

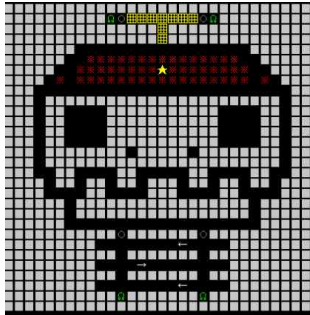
이후에 화살에 대한 처리는 플레이어가 못 가도록 다른 형태의 물체로 지정(11.7) -> (오류가 많아서, 사용하지 않아서) 그냥 화살과 공을 같이 상쇄하는 처리(11.9)

11월 7일에는 새로운 스테이지를 구상하고, 기존에 있던 부분을 고쳤다. (Z 모양으로 1234)



데이터를 포인터를 이용하다가 전역변수를 사용하니, 속도의 차이가 굉장했다. 이후에 if문을 switch문으로 교체하여 효율적으로 데이터를 이용하게 되었다.⁶⁾

11월 10일 스테이지 6 추가 & 각 스테이지에 다른 색깔 적용.



사라진 아이디어

- a. 시간을 입력받아 랭크 기능 구현
- b. 사용자로부터 맵 받아오기

(복잡함)

11월 11일 메인메뉴 추가 (표지그림 참조) ‘▶’를 통한 메뉴 선택 기능 구현.

5. FMOD를 이용한 소리 출력

게임을 만들면서 다른 사람이 만든 게임을 보게 되었다. 특별한 점은 느껴지지 않는 게임이었는데, 찾는 사람이 많았다. 그 이유를 곰곰이 생각해보니 소리를 정말 잘 구현했다. 효과음이 게임을 한 차원 높은 수준으로 끌어올린 것이다. 저번에 만들던 게임에서 구현을 시도하다가 복잡해서 그 만됐던 ‘소리’를 이번에는 꼭 넣고 싶었다. 11월 12일 약 5시간 동안 찾아다니던 중에 드디어 FMOD를 이용해서 처음으로 소리를 출력하는 데 성공했다. 이날은 많은 기능을 추가하지는 않았지만, FMOD의 느낌을 조금 이해했다는 것에 의미를 두었다.



스피커 기능 구현 (타이틀 노래만) - 11.12

11월 13일 플레이어 움직이는 소리, 폭탄 터지는 소리, 폭탄 설치 소리 등 소리를 추가했다.

14일 과자블록 터지는 소리. 스테이지 clear 소리, 죽음 소리, 잡다한 예외처리.

11월 15일 대규모 사운드 공사

메인메뉴 배경음악만 스피커 기능을 구현에서 모든 사운드에 동일 하게 적용되는 스피커 기능이 필요했다. 따라서 기본적으로 소리가 큰 경우에 조정이 필요했고 무료 데시벨 조절 사이트에서

6) if문이라면, 속도 차이로 게임이 만들어지지 않았을 것이다.

소리 조절 작업 후 이것을 토대로 적용했다.

이후에 배경음악과 효과음이 동시에 켜지지 않는 오류가 발생하여 배경음악을 전역변수로 적용하여 해결. (채널 수의 문제도 포함됨 OR update 처리를 하지 않음)

11월 15일

death(죽은 횟수)가 적은 순으로 정렬하여 랭크 기능을 구현하고, 이렇게 정렬한 데이터를 메모장에 저장하는 것을 구현하는 데에 애를 먹었다. 해결하지 못한 상태로 일단 대기.

랭크 선택(반복적) 시에 이상 현상 발생을 인식하고 해결을 위해 노력했으나 지금(11.24)까지도 성공하지 못했다.⁷⁾

화살 - 공 - 플레이어 사이에서 화살과 공의 위치가 겹칠 때 사라지지 않는 현상 발생.⁸⁾

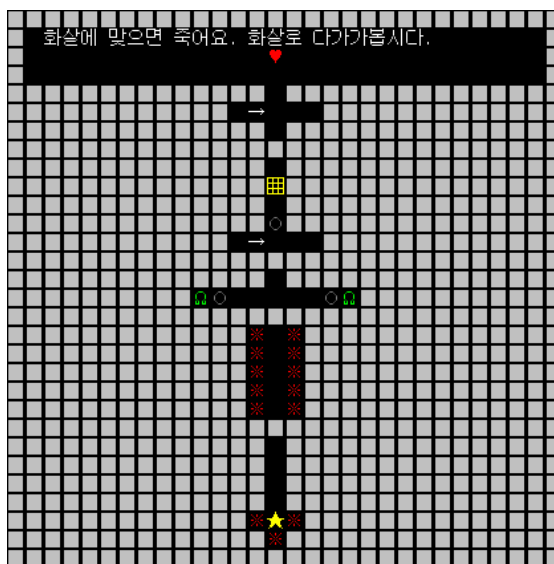
화살이 죽은 경우에(공과 겹침) 그 자리에 폭탄을 설치할 시 다시 살아나는 현상 해결.

화살이 죽은 경우에 좌표를 (0,0)을 데입하고 이후에 실행되는 경우 (arrowMove에서) 넘기는 것으로 처리.

플레이어 - 공 - 과자블록

이런 형태의 상황일 때 플레이어가 공의 위치로 갈 수 있는 오류를 해결.

(비슷한 경우를 포함하여)



튜토리얼 기능을 추가.

작은 퀘스트들로 이루어져 있으며

조건을 만족하면 다음으로 넘어감...

플레이 중에 사용하는 기술은 모두 설명하고 있다.

7) 고의적으로 랭크 메뉴를 선택하는 것이 아니면 문제가 없다.

8) 11.24일 문서화 중에 처리.

11월 17일

재귀함수에서 꼬이는 loop를 해결하여 그 정도를 줄였다.

튜토리얼 중에 'm'(main menu) 입력 시 오류가 발생하여 제거.



저장된 데이터



랭킹 기능 실행 시

랭킹 기능의 오류 해결. name[0]을 통해서 이름이 있는지를 검사했다.

스테이지 선택 기능에서 loop 꼬임 현상 발견

11월 18일

잡다한 문제(주로 loop꼬임 현상)에 대해 처리하고 소스코드 문서화 작업 진행.

fmod.dll 파일, 사운드 파일, 실행 파일을 정리

11월 19일

컴퓨터의 성능에 따라 화살의 속도를 조절할 수 있는 기능을 추가.⁹⁾

11월 22일

방향키에 따른 오류를 최대한 제거, 화살의 속도 조절을 게임 중에 설정할 수 있도록 변경.

11월 24일

튜토리얼에서 'M : 메인메뉴' 가 보이지 않도록 설정.

9) 속도가 느려지는 주요 원인은 gotoxy() 때문인데, 이건 건드릴 수 없으니 Sleep의 인자를 변경한다.

6. Text Maze의 주요 기능

가. 게임 플레이

- 1) 폭탄 설치 / 터트리기 (과자블록)
- 2) 공 밀기 / 화살 없애기
- 3) 슬로우 블록
- 4) 스위치를 통한 블록 사라짐 효과
- 5) 플레이어 사망처리
- 6) ‘m’, ‘M’ 입력 시 메인메뉴로 이동
- 7) 도착점에 대한 처리
- 8) ‘1’, ‘2’, ‘3’ 키를 이용한 화살 속도 조절
- 9) 좌우상하 이동 처리 WASD
- 10) BackSpace 스테이지 재시작
- 11) ESC 입력 시 게임 종료

나. 소리

- 1) 각 스테이지에 따른 노래 설정
- 2) 스피커 기능
- 3) ‘T’, ‘t’ 입력 시 효과음 ON/OFF
- 4) ‘스테이지 선택’ 노래 미리 듣기

다. 사용자 인터페이스

- 1) ‘▶’ 문자를 통한 메뉴 선택 (타이틀 / 스테이지 선택)
- 2) ‘스테이지 선택’에서 스테이지 미리보기 기능

라. 파일 입출력

- 1) 모든 스테이지 클리어 시 자신의 이름을 적을 수 있도록 하여 랭킹 데이터로 활용
- 2) 기존에 입력된 데이터와 새로 입력된 데이터를 함께 정렬하여 랭킹 기능 구현

7. 프로그램 흐름도

메인메뉴로 이동 / 다음 스테이지 이동 / 스테이지 선택 후 처리 등 구조적인 것만 다룬다.

함수 내부는 생략 ○ 기호는 루프 내부를 통한 반복

프로그램의 초기 loop 이전

```
system("mode con cols=100 lines=34 "); //CLI 크기
system("title TEXT MAZE");//상단 이름

/* 커서 없애기 */
CONSOLE_CURSOR_INFO ConsoleCursor;
ConsoleCursor.bVisible = false;
ConsoleCursor.dwSize = 1;

SetConsoleCursorInfo(GetStdHandle(STD_OUTPUT_HANDLE), &ConsoleCursor);

stage = -1;

SetMaze(); // 그니까 구체적으로 표현하면 여기서는 인자를 따로 받지 않지만, 다른 코드에서 온갖 데이터 처리는 모두 한다.
//미로 설정 title이 내부에서 시작되어 키값을 받고, 이후에 stage값을 변동.

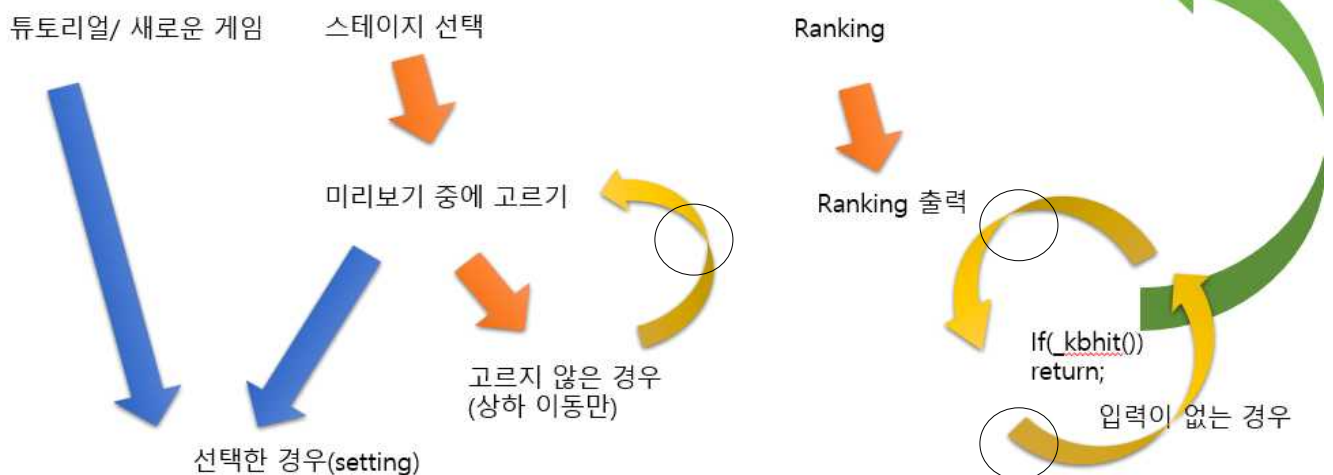
SetMaze(); // 바뀐 stage값을 이용하여 다시 셋팅.

char cInput = NULL;
Fmod->update();

while (_kbhit()) _getch();

while (true)
{
```

(AA) SetMaze에서는 title을 호출 (파일 입출력 처리)



초기값을 토대로 다시 SetMaze 함수 호출 → 이후에 게임 플레이 loop

설정한 대로 게임 시작 (SetMaze이후 출력 부분)

```
while (true)
{
    OutPut();
    //미로 출력

    //개발 함수
    if(create)
        playerPos();

    // tEndPos == tPlayerPos 에 대한 j
    player Inform(); // 플레이어 inform

    if (_kbhit())
    {
        cInput = _getch();

        while (_kbhit()) _getch();
    }
}
```

```
switch (cInput)
{
    case '1':
    case '2':
    case '3':
        arrowSpeedSet = cInput - '0';
        speedShow(); // 바뀐 값 보여줌
        speedSetting(); // 값이 변하면

        break; // 이후 셋팅 값에 따라

    case ESC:
        exit(0);
        break;

    case SPACEBAR:
        if(stage != 0 || quest >= 2)
            CreateBomb();
        break;
}
```

가장 큰 이유 :
openDoor();

화살의 수/ 스위치 등을
세어준다. 좌표 잡기

```
case 'q':
case 'Q':
    Fire();
    break;
case BACKSPACE: //
    SetMaze(); //
    OutPut();
    if(stage != 0)
        death++; // t
    break;
case '=':
    //+
    speakerPlus();
    break;
case '-':
    speakerMinus();
    break;
case 't':
case 'T':
    mute();
    break;
case 'g':
    status();
    break;
// ...
```

해당 스테이지
다시 시작.

```
case 'm':
case 'M':
    if (stage == 0) break; // tutorial에서 m을 막음
    //하위 요소 해제
    music->release();
    // 상위 요소 해제.
    Fmod->release();

    stage = -1;

    SetMaze(); // title에서 stage 설정.
    SetMaze(); // 다음 셋팅에서 적용.

    OutPut();

    while (_kbhit()) _getch();

    break;
}

MovePlayer(cInput);
arrowMove();
openDoor();
// 스위치 개수에 따른 처리.
}
else
    arrowMove();
```

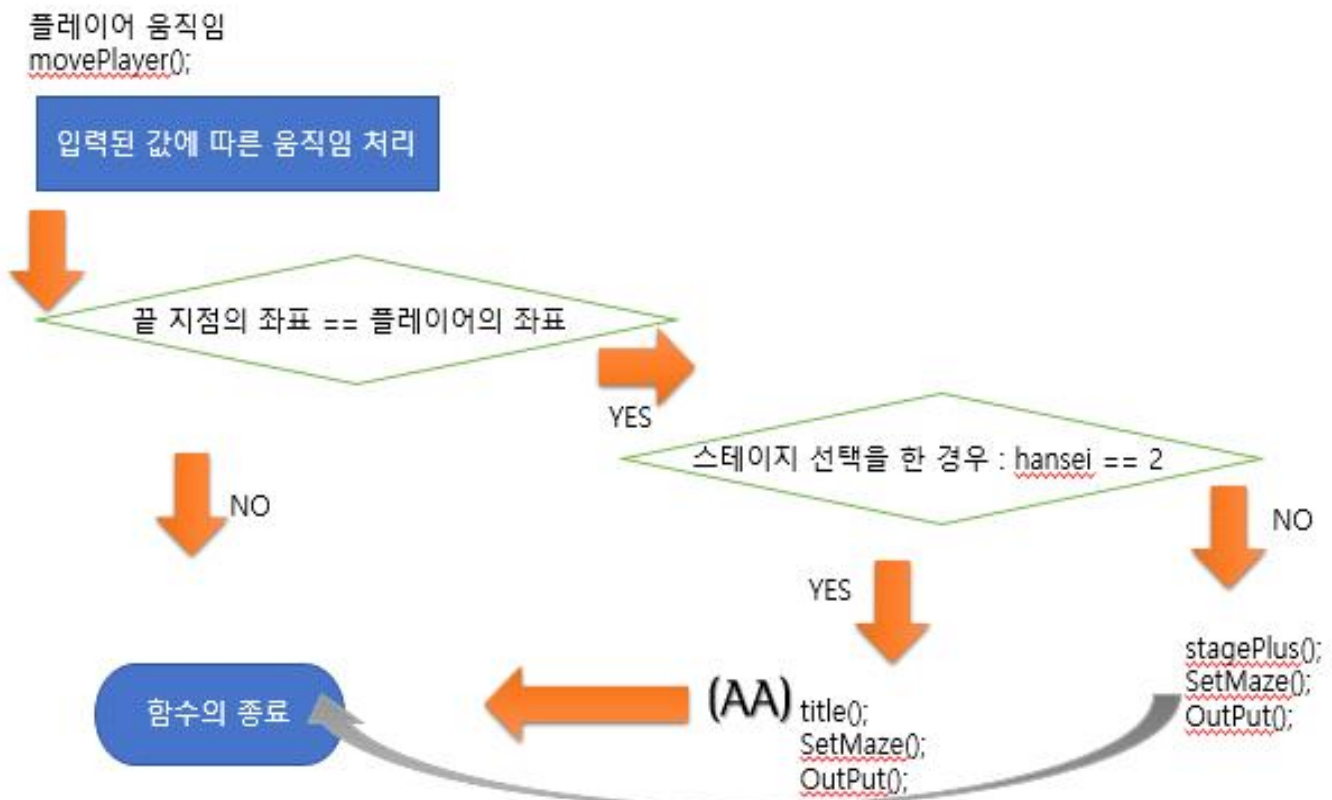
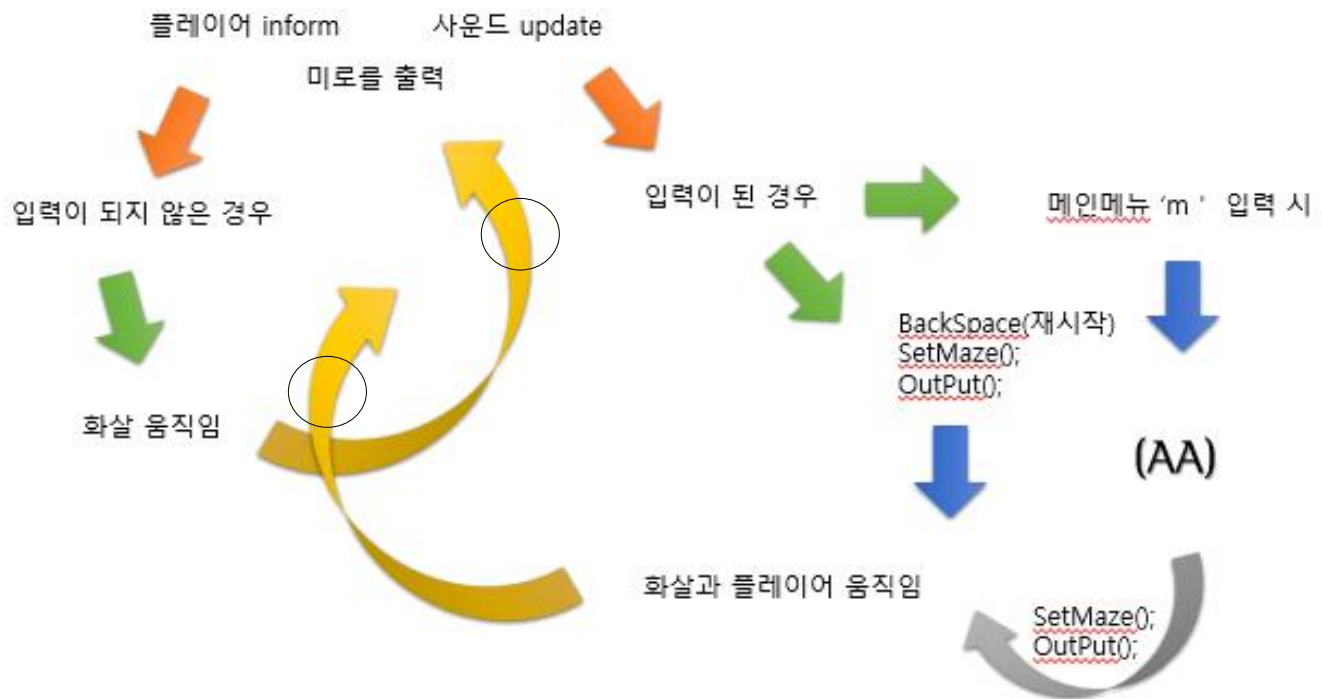
```
if (stage == 0) // tutorial
    tutorial(cInput);
//일단 튜토리얼에서 맵이 뜨면 음:

channel1->setVolume(speaker); //
Fmod->update();
```

}

```
system("pause");
return 0;
```

}



8. 수정하지 못한 ERROR

1) 특수키 (← ↑ → ↓) 입력 시 'm'을 누른 것으로 처리가 되는 오류¹⁰⁾

처리 : 특수키를 입력받지 않는 것으로 처리

2) title()에서 랭킹을 반복적으로 선택하는 경우(꼬리에 꼬리를 물고 함수를 호출)

처리 X : 고의적인 반복이 아니면, 오류는 발생하지 않는다. 어느 정도 선택을 반복하면 튜토리얼이 실행이 되고 이후부터 소리는 나오지 않는다. 스테이지(1 stage)가 나오기는 하나, 소리가 제대로 출력되지 않고 이 상태로 'm'을 누르게 되면 메인메뉴가 아닌 다른 모습의 스테이지가 보인다.¹¹⁾ 함수의 반복을 끊어주기 위한 처리를 시도했으나, 결국 제대로 실행이 되지 않아 패스.

9. 기타사항



게임의 아이콘으로 사용한 이미지다.

Pixabay 무료 이미지를 편집하여 사용

cheat와 배포버전의 차이점 : 함수는 동일하나, '0' (cheat key)에 대한 처리에서 차이가 있다.

'q' status key

FMOD 레퍼런스

<http://blog.naver.com/PostList.nhn?blogId=sayzy&from=postList&categoryNo=38>

https://www.youtube.com/watch?v=u_PZhBmhHrY

10) 초기에는 문제가 없었으나 코드가 길어지면서부터 인식이 안 된 것으로 추정한다.

11) stage == -1일 때 벽('0')을 ◆문자로 출력한다.