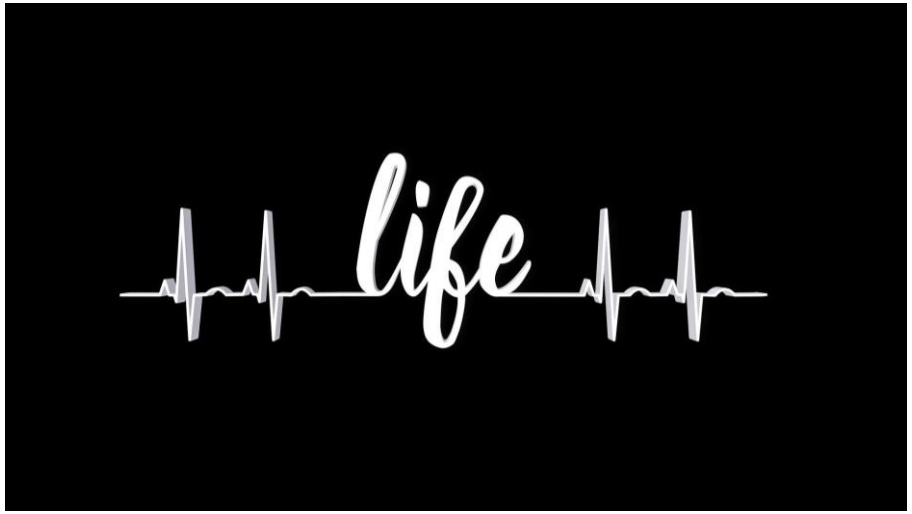
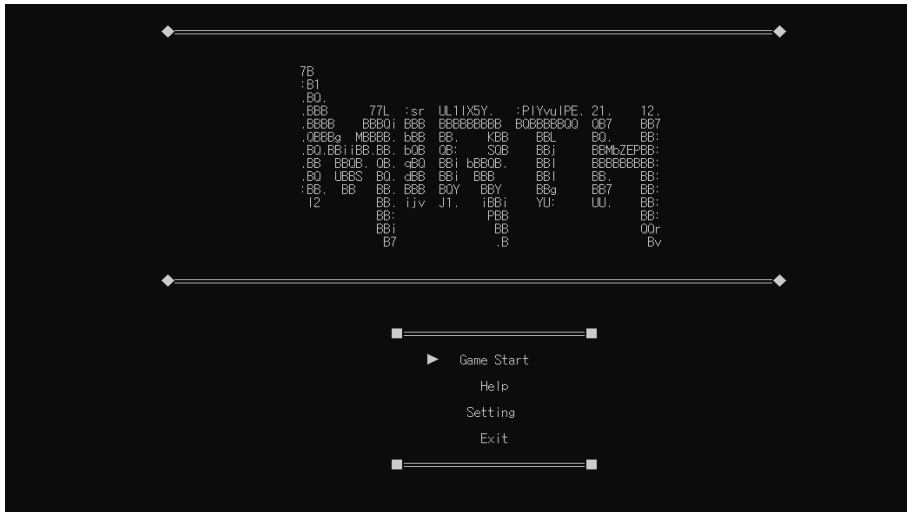




Mirth

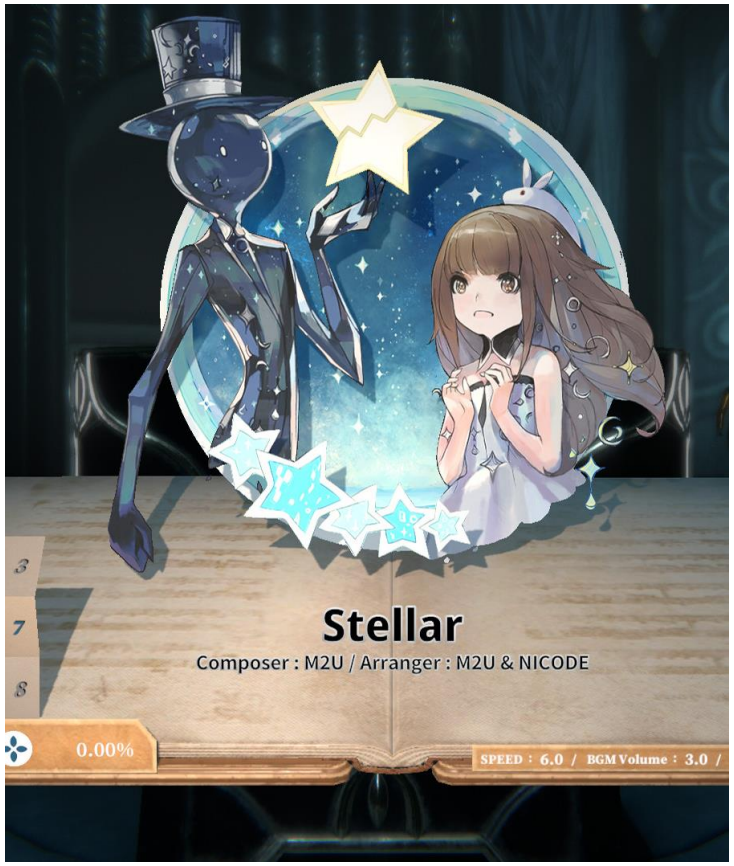
김주영

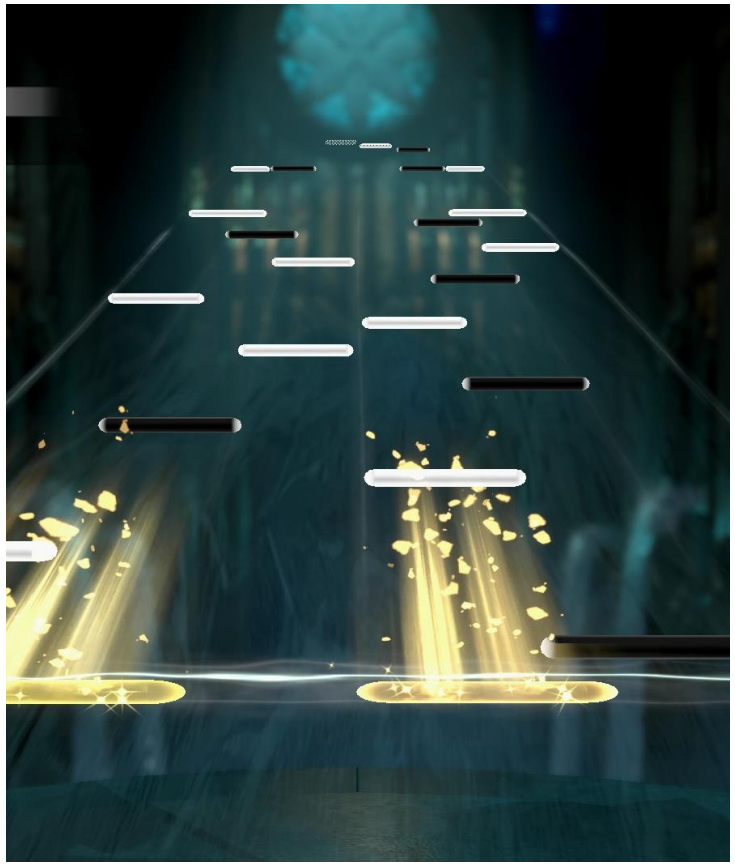
219124116

컴퓨터공학부

컴퓨터소프트웨어전공

게임 개요: 리듬게임

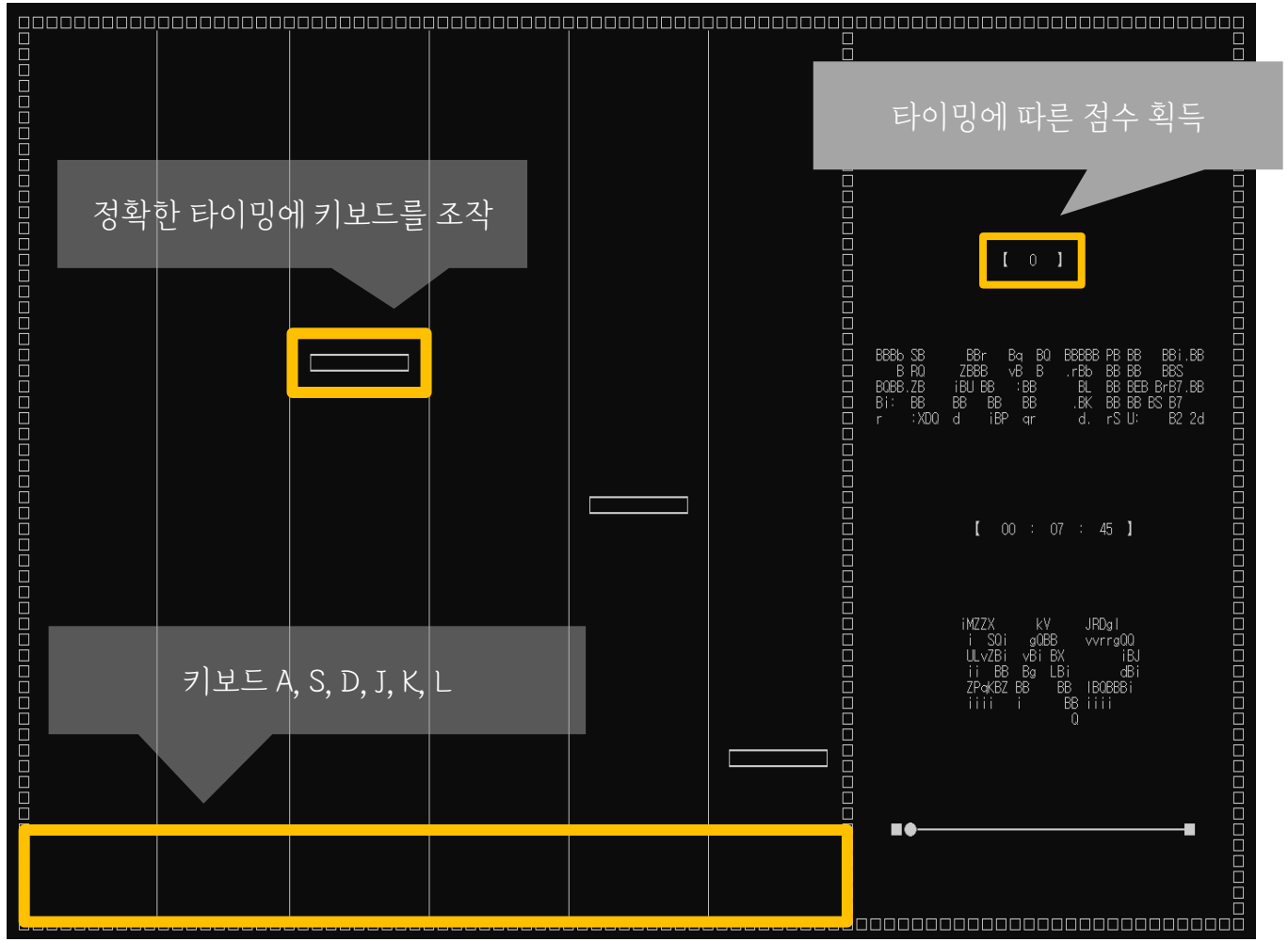


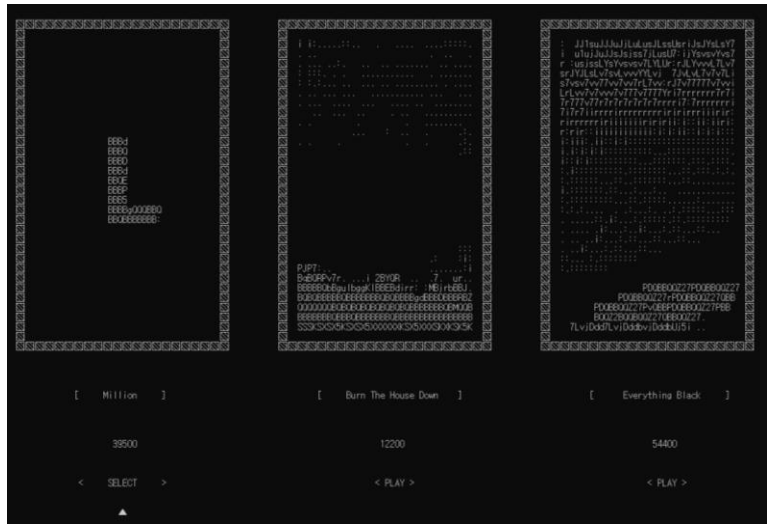


✓게임 설명

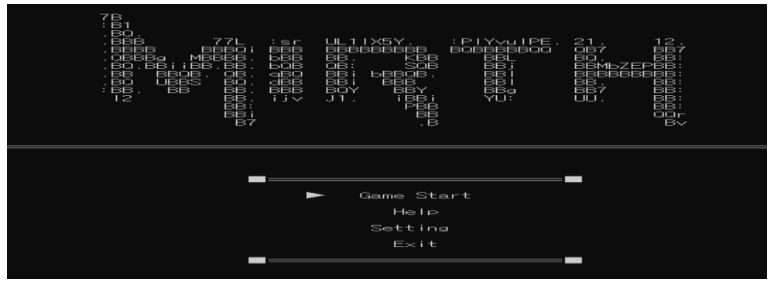
- 게임 플레이 음악에 맞춰서 정확한 타이밍에 지정된 조작을 행하는 게임 장르이다.
- 상단에서 내려오는 바의 타이밍에 맞춰서 키를 누르면 점수를 얻으며 높은 점수를 목표로 하는 게임이다.

게임 스펙: 주요 화면





수록 곡 선택 및 점수 확인 화면



메인 메뉴 조작, 도움말, 설정 화면

게임 스펙: 요구 사항

- 소프트웨어
  - 64비트 ARM을 지원하며 유니코드(UTF-8)을 지원하는 운영체제
  - C++ 표준 라이브러리
  - FMOD 사운드 라이브러리

- 하드웨어
  - FHD(1920x1080) 해상도를 지원하는 모니터
  - 2채널 이상의 스피커, 풀영문 자판 키보드
  - 1GB 이상의 FHD를 지원하는 비디오 그래픽 카드
  - 쿼드코어 이상의 CPU, 4GB 이상의 RAM, 20GB 이상의 디스크 공간

게임 설계

리듬 게임의 중요 요소는 음악과 박자이다.

Fmod Library를 사용하면 소리를 두 개 이상 동시에

출력 가능 하므로 배경음과 효과음을 구현할 수 있다.

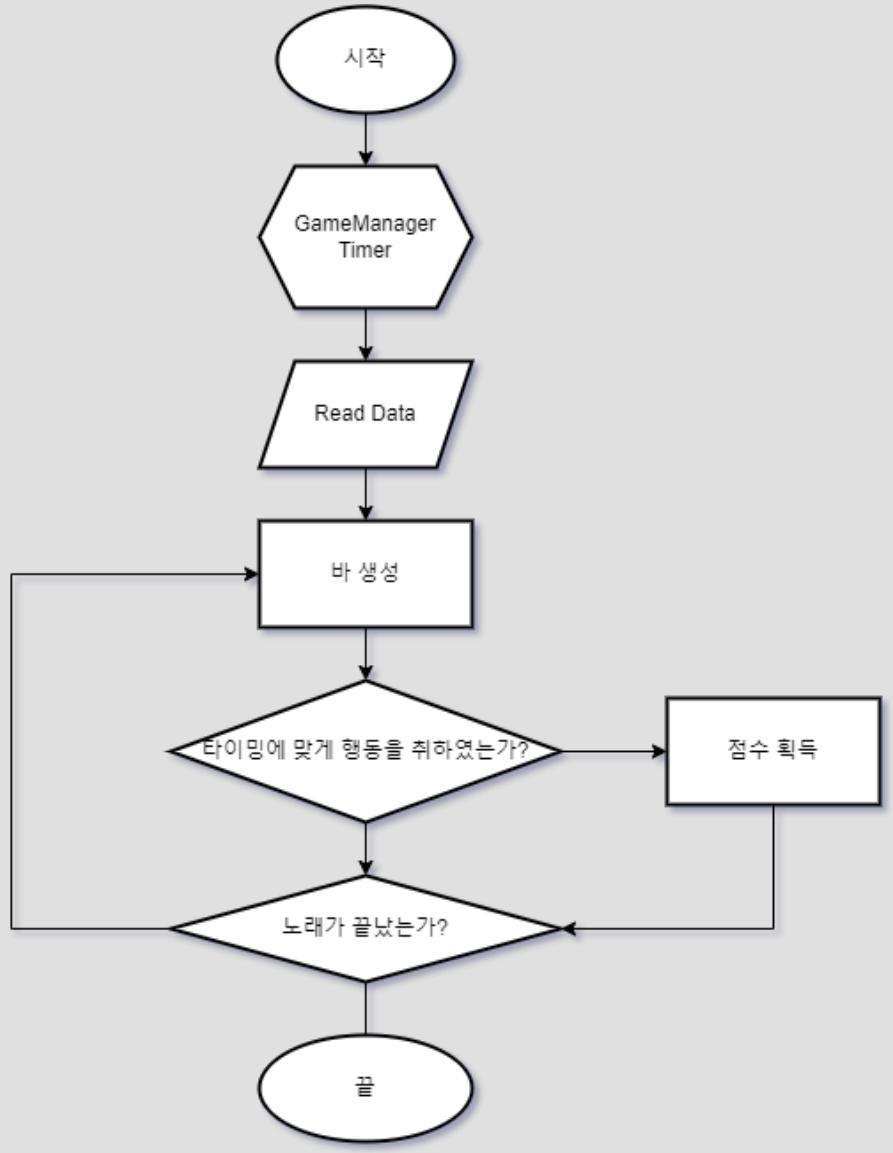
그리고 노래 박자에 맞게 분, 초, 밀리초가 적혀 있는 데이터 파일을

만들어서 게임 플레이 타이머에 맞게 바를 생성하면 박자에 맞는

바가 생성되어 내려올 수 있는 환경을 구현할 수 있다.

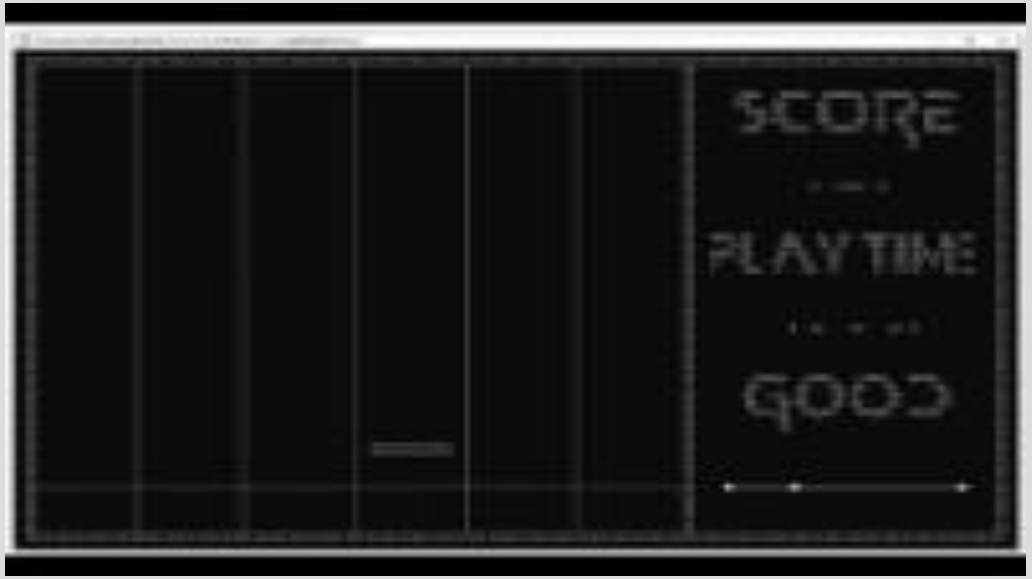
GameManager와 SoundManager를 이용하면 게임 환경에

쉽게 접근할 수 있도록 만들 수 있다.



데모 (Demonstration)

https://youtu.be/GZHb-o9NwnE



< 유튜브 영상 >

< 미리보기 사진 >

주요 Source Code

```
class GameManager
{
private:
    int end_m_time, end_s_time, end_ms_time;
public:
    array<bool, 6> keeyboard_flag;
    string ver = "1.0.0";
    int room_number;
    string play_time, play_score;
    double time, end_time;
    int m_time, s_time, ms_time;
    int state, now_score;
    int bg_sound, eff_sound, sync;
    vector<vector<string>> note[3];
    vector<vector<string>> setting;
    vector<vector<string>> score;
    vector<vector<string>> read_text(const string);
    GameManager();
    string get_play_time();
    string get_now_score();
    void set_end_music_time(int, int , int);
    void set_end_time();
    void set_time();
};
```

```
vector<vector<string>> GameManager::read_text(const string uri)
{
    string ch;
    string ch_tmp;
    vector<vector<string>> vec;
    ifstream textf(uri);
    if (textf.is_open())
    {
        vector<string> tmp;
        while (!textf.eof())
        {
            ch = textf.get();
            if (ch == ";")
            {
                vec.push_back(tmp);
                tmp.clear();
                continue;
            }
            else if (ch != " " && ch != "\n" && ch != "\0")
            {
                if (ch != ",")
                {
                    ch_tmp.append(ch);
                }
                else
                {
                    tmp.push_back(ch_tmp);
                    ch_tmp = "";
                }
            }
        }
        textf.close();
    }
    else
    {
        cout << "파일을 열 수 없습니다." << endl;
    }
    return vec;
}
```

GameManager 클래스와 read_text 메소드이다.

게임의 핵심인 data 파일(타이밍, 사용자 설정, 점수)을

불러와야 하는데 text 파일로 되어있는 text 파일을 읽어서

vector로 변환하여 반환하는 핵심 코드이다.

```
// 내려오는 노트들의 상태 체크 및 내려오는 코드
// 키보드 눌렀을 때 상태도 체크 함
for (int i = 0; i < note_amount; i++)
{
    // 노트(바)들을 미리 정의된 시간에 맞춰서 생성 하도록 함
    // 밀리초 부분을 ==으로 하면 생성조차 안되는 것이 많음
    if (bar[i].m_time == gm->m_time && bar[i].s_time == gm->s_time &&
        to_string(bar[i].ms_time)[0] == to_string(gm->ms_time)[0] &&
        to_string(bar[i].ms_time)[1] >= to_string(gm->ms_time)[1])
    {
        bar[i].create_flag = true;
        bar[i].flag = true;
    }

    if (bar[i].create_flag == true)
    {
        // 노트(바)들이 밑으로 내려가도록
        vector<int> vec = bar[i].get_move_bar({ bar[i].now_coord[0], bar[i].now_coord[1] });
        ScreenPrint(vec[0], vec[1] - 1, bar[i].sprite[0]);
        ScreenPrint(vec[0], vec[1], bar[i].sprite[1]);

        // 노트들이 끝에 닿으면 제거
        if (bar[i].now_coord[1] >= 57)
        {
            bar[i].state = 3;
            bar[i].create_flag = false;

            gm->state = 1;
        }
```

play_room 함수안에 구현 되어있는 Bar를 좌표에 맞게

출력하는 부분이다. Bar 객체에 저장 되어있는 타이밍이

현재 타이밍과 일치하면 좌표에 맞게 화면에 출력된다.

Bar의 좌표는 끝에 닿기 까지 계속 하락한다.

✓게임을 만들면서 어려웠던 점

콘솔 게임을 전부터 만들어 보고 싶었지만
막상 해보니까 유니티로 게임을 만드는 것보다
훨씬 매우 까다로웠고 직접 구현해야 하는 것이 많아서
시간도 많이 걸리고 오류도 많이 생겼습니다.

그리고 리듬 게임 관련해서 인터넷을 찾아봤지만
제 코드와 비슷하게 구동되는 코드가 아니었습니다.
더블 버퍼링 말고 리듬게임 관련해서는
참고를 할 만한 소스가 없었고 심지어
주변에도 없었기에 오로지 제 힘으로만
코딩을 해야 했던 부분이 어려웠습니다.

✓현재 고민 중인 부분

현재 3가지 버그가 있습니다.
바 생성이 정해진 타이밍을 벗어나 이상하게 생성되는 점, 키보드 6개의 버튼을 동시에 클릭하면 음악이 재생되다가 멈추는 점, EXE 파일로 만들고 실행했을 때 갑자기 종료되거나 제대로 실행이 안되는 점
이렇게 3가지를 해결해야 합니다.

일단 바 생성 부분은 분, 초 이렇게 정해 놓았던
타이밍 대신 바가 생성되고 몇 초 뒤에 생성 되도록
그 몇 초를 대신 적어 놓는 방식으로 바꾸는 대안을
생각해 두었습니다. 하지만 나머지 버그는
조금 더 고민을 해봐야 할 것 같습니다.

✓게임의 장단점

일단 다른 콘솔 리듬 게임과 비교했을 때 제 게임이 디자인적 요소에서 아주 조금 더 장점이 있다고 생각이 됩니다.유튜브에 검색한 다른 리듬게임은 ■■■ 이렇게 바를 만들었지만 저는 ┌──┐ , └──┘ 이 두 개를 써서 바를 만들었기 때문에 시각적으로 장점이 있습니다.

다른 콘솔 게임과 비교했을 때는 콘텐츠가 3개의 노래 밖에 없다는 단점이 있지만 언제든 추가할 수 있고 데이터 저장 기능이 있다는 것이 장점이라고 생각됩니다.