자료구조 프로젝트

<MP3 음악 플레이어>

**음악 플레이어**

이 프로젝트에서는 그래픽 렌더링 라이브러리 및 openGL, Win32 등의 기능을 활용하여 mp3 파일을 관리하고 재생하는 음악 플레이어를 개발하고자 한다. 위에서 언급한 특정 기능들의 의존성으로 인해 Windows에서만 컴파일 및 작동하며, 개발 환경은 Windows 7 및 Windows 10이다.

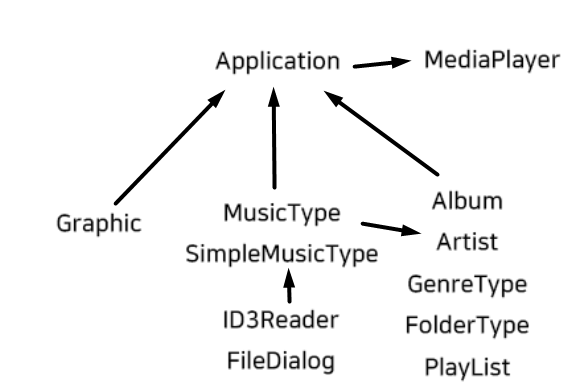
**기능**

1. 곡, 앨범, 아티스트 정보를 관리할 수 있다.
2. 곡을 선택하여 재생할 수 있다.
3. 최근 추가한/가장 많이 재생한 음악 목록을 볼 수 있다.
4. 특정 곡 (혹은 앨범, 아티스트) 목록을 기반으로 플레이 리스트를 생성할 수 있다.
5. 위에서 생성한 플레이 리스트를 재생목록 파일로 출력할 수 있다.
6. 로컬 폴더에서 파일 목록을 읽어 음악 데이터를 생성할 수 있다.
7. Mp3 파일에서 ID3 태그를 읽어 곡 정보에 대입할 수 있다.
8. GUI를 활용하여 데이터 관리를 직관적으로 파악할 수 있다.
9. 그 외에 장르 및 폴더 별로 음악 목록을 확인할 수 있다.
10. 사용자가 특정 필터를 조합하여 원하는 음악 목록을 검색할 수 있다.

**시나리오**

사용자는 파일 또는 폴더를 통해 음악 파일들을 읽어와 음악 데이터를 추가할 수 있다. ID3 태그에서 읽어온 정보, 혹은 사용자가 입력한 정보를 바탕으로 앨범/아티스트/장르/폴더 별로 음악을 자동 분류하여 보여준다. 또한 화면에 출력된 음악 목록들을 하나의 플레이 리스트로 변환하여 저장할 수 있다. 이 플레이 리스트는 플레이어에서 재생할 때 순서대로 재생할 수 있으며, 이를 wpl/zpl/m3u8 파일 형태로 출력하여 실제 프로그램 (Window Media Player, Groove & 음악, 곰오디오 등)에서 재생할 수 있다.

**ADT**



이 프로그램에서 클래스 연결도는 위 그림과 같다.

ID3Reader나 FileDialog를 통해 MusicType을 읽어올 수 있고, 이는 각 앨범/아티스트 등의 데이터에 자동으로 추가된다. 이에 관련한 Graphic 객체를 생성하여 화면에 렌더링하고, 이 모든 과정을 Application에서 관리한다. 또한 Application에서는 MediaPlayer와의 상호작용을 통해 음악을 재생할 수 있다.

이 명세서에서는 Application의 구조만 간략하게 설명한다. 추후 실제 개발에서 변경될 수 있다.

**Class Application**

<Member variables>

std::ifstream m\_inputFile; ///< 파일 입력을 받기 위한 스트림

std::ofstream m\_outFile; ///< 파일 출력을 하기 위한 스트림

BST<MusicType> musicList; ///< 모든 음악을 저장하는 리스트 (master list)

BST<SimpleMusicType> nameList; ///< 이름별로 저장하는 리스트

BST <GenreType> genreList; ///< 장르를 저장하는 리스트

CircularQueueType<MusicType> newAddMusicList; ///< 최근 추가된 음악 리스트.

BST <Album> albumList; ///< 앨범 리스트

BST<Artist> artistList; ///< 아티스트 리스트

LinkedList<MusicType> mostPlayedList; ///< 가장 많이 재생한 음악 리스트

MediaPlayer player; ///< 음악 플레이어

<Member functions>

void AddMusic();

void DeleteMusic();

void ReplaceMusic();

void DisplayAllMusic();

void AddPlayList();

void DeletePlayList();

void ReplacePlayList();

void PlayMusic();

void NextMusic();

void PrevMusic();

void PauseMusic();

void WriteDataToFile();

void ReadDataFromFile();

void Search();