

2차 창작기능과 사용자 자유도개선을 제공하는 음악 플레이어의 개발

이희준† · 김진관† †
동안고등학교† · 백양고등학교† †

Development of a music player that provides fanwork-design system and professional player functions easily for users

Heejun Lee† · Jinkwan Kim† †
Dong-an High School† · Paik-yang High School† †

요 약

본 연구에서 제작한 음악 플레이어인 Symphony(이하 심포니)는 사용자의 음악적인 공감을 돕고, 그에 따른 음악을 기반으로 한 2차 창작의 접근성과 자유도를 높여주는 기능을 포함하는 복합적인 프로그램이다. 기존 플레이어는 음원의 유통과 선별적 재생에만 초점을 두는 경향이 있는데, 심포니는 음악 플레이어의 개념을 사용자 입장에서 확장하여 가사 시스템, 춤 시스템, 플레이어 스킨, 사운드 비주얼라이저, DSP 설정 및 실시간 DSP 스크립팅 등을 추가하거나 개선하였다.

주제어 : 음악 플레이어, 2차 창작, 2D/3D 애니메이션, DSP, 사용자 경험

1. 서 론

멜론, 벅스, 지니 등의 대형 음원 서비스 플랫폼들은 현재 독자적인 음악 플레이어를 소비자에게 제공하고 있다. 이러한 음악 플레이어는 음원 유통과 곡의 선별적 재생에만 주 초점을 두는 경향이 있다. 소비자는 이에 따라 좋아하는 음원을 재생하고 음악 플레이어를 끈 뒤, 자신들의 일상을 이어나간다. 평범해 보이지만, 이는 음악 플레이어를 과소평가하는 행위이다.

음악의 세계에서는 매일 어마어마한 변화가 일어난다. 이와 더불어 음악가는 자신의 음악을 최고로 표현하기 위해 연습에 막대한 시간을 할애한다. 하지만 청자에게는 음악가의 이러한 정성에 진심으로 공감하고 즐길 수 있는 수단이 그다지 없다. 또한, 음악을 듣고 떠오르는 무한하고 즐거운 상상을 청자가 직접 쉽게 표현할 수 있는 수단 또한 애석하게도 그다지 없다.

우리는 음악 플레이어가 이러한 점을 해결할 수 있는 가능성을 지닌다고 생각한다. 그래서 심포니는 음악을 사랑하는 모든 사람들에게 쉽고, 아름답고, 체계적인 놀라운 형태의 워크스테이션을 제공한다. 더불어, 사용자의 환경에서 음원을 최고의 느낌으로 청취할 수 있도록 논리를 이어나가고 있다.

2. 배경

2.1 관련 프로그램

관련된 프로그램으로써 알송과 곱오디오가 존재한다. 알송은 이스트소프트사에서 개발한 음악 플레이어이다[1]. 독자적인 가사 시스템을 보유하고 있는 것이 특징으로, 이를 이용한 유저들의 다양한 창작활동(실시간 자막 아트 등)이 이루어졌다. 사용자는 가사를 텍스트 형식으로 각 가사의 줄마다 싱크를 조정하여 가사를 제작할 수 있다. 심포니의 전문적 가사 에디터에 영향을 주었다. 또 다른 관련 프로그램인 곱오디오는 Gom & Company 에서 개발한 음악 플레이어이다[2]. 알송과 유사한 가사 시스템을 가지고 있고, 플레이어의 외관을 바꿀 수 있는 스킨 시스템을 가지고 있다.

2.2 심포니의 기반 기술

심포니는 다양한 기술을 기반으로 개발이 되었다. 사용 언어는 C#으로 높은 생산성을 기반으로 빠르고 정확한 구현을 가능하게 했다. 그리고 UI 툴킷과 렌더링을 위해 WPF와 DirectX를 사용하여 미려한 UI와 가볍고 빠른 3D렌더링을 가능케 하였고 DSP의 실시간 스크립팅을 위해 가볍고 간략한 Lua 언어를 사용했다. 가사와 같은 사용자의 창작물을 사용자간 공유

를 구현하기 위해 PHP와 MySQL을 이용하여 데이터 서버를 구현하였다. 이하는 각 기술에 대한 간략한 설명이다.

사용된 언어인 C#은 Microsoft사에서 개발한 객체 지향적 프로그래밍 언어이다[3]. C#은 JVM과 유사하게 작성된 코드를 바이트 코드로 컴파일 하여 저장하고, 실행 시 JIT을 통해 네이티브 바이너리로 컴파일 하여 실행하는 구조를 갖고 있다. C#은 윈도우즈와 친화적이고 풍부한 기본 프레임 워크와 SQL을 코드에서 실행해주는 LINQ등의 기술들을 기반으로 한 뛰어난 생산성이 특징이다.

UI 킷으로써 사용된 WPF(Windows Presentation Foundation)은 Microsoft사에서 개발한 벡터 그래픽 기반의 Windows용 UI Toolkit(TK)으로 기본적인 UI 레이아웃, 이벤트 처리를 가지고 있고, 데이터 컨텍스트를 통해 UI에 값이나 명령을 전달하는 데이터 바인딩을 지원하는 것이 특징이다[4]. DirectX를 통한 렌더링 시스템으로 기존의 윈도우의 UI TK인 WinForm 대비 빠른 성능을 보인다.

3D렌더링을 위해 사용된 DirectX는 Microsoft사에서 개발한 그래픽스, 미디어 인코딩/디코딩 등의 기술들을 포함한 라이브러리이다. DirectX에는 3D와 2D렌더링을 위한 Direct3D, Direct2D패키지가 존재하고, 오디오, 비디오 인코딩 디코딩을 위한 DirectShow와 XAudio등이 존재하고 있다.

실시간 스크립팅을 위해 사용된 Lua는 스크립트 언어로써, C계열의 언어에서 자주 사용된다[5]. 간단한 문법과 가벼운 인터프리터 구현을 기반으로 실시간 처리에서 높은 성능을 낼 수 있다. 또한 언어 구조적으로 C와의 공존이 손쉽도록 설계되어있는 언어이다.

3. 심포니의 기능

심포니는 기존 음악 재생 프로그램의 무미건조함과, 음악에 관한 2차 창작의 어려움을 개선하는 데에 중점을 둔 음악 청취 프로그램이다. 심포니는 기존의 음악 청취 프로그램과는 다르게 음악을 재생할 뿐만 아니라 사용자가 음악에 더 몰입할 수 있도록 하고, 음악과 관련한 2차 창작을 편리하게 할 수 있도록 한다.

심포니는 사용자에게 수려한 가사 시스템을 제공한다. 가사는 노래에 의미를 부여하기에 가장 직접적이고 강한 수단이다. 이러한 가사에 사용자는 내장 에디터를 통해 애니메이션을 입히고, 텍스트 폰트와 크기 등을 조정하고, 색과 그림자, 흐리게 이펙트 등을 설정할 수 있다. 또한 고급 기능으로써 세세한 애니메이션 KeySpline, 지능적인 가사 출력을 위한 내용의 화면 출력 레이아웃 기능들을 구현함으로써 여러 가사 오브젝트끼리 조합하여 애니메이션을 제작할 수 있게 환경을 구성하였다. 그리고 자유로운 연출을 위하여 이미지 오브젝트를 지원한다.

심포니는 사용자에게 강력한 DSP 기능과 설정들을

제공한다. Lua를 통해 사용자는 실시간으로 DSP의 제작과 시연을 할 수 있고, DSP 체인을 제작하여 사용자가 직접 소리 효과를 구성하고 체인 순서를 창의적으로 변경할 수 있도록 했다. 이러한 기능으로 사용자는 기존보다 더 자유롭게 음악을 자신의 스타일에 맞춰 들을 수 있다. 또한 사용자는 여러 DSP를 실시간으로 동시에 변경하여 자신의 생각대로 청취 환경을 설정할 수 있다.

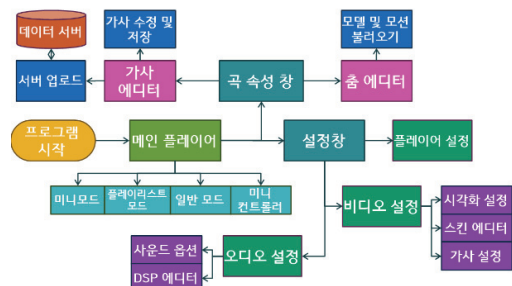
심포니는 기존의 주요한 플레이어들이 제공하지 않는 참신한 댄스 시스템을 제공한다. 노래에 맞추어 추는 춤은 그 자체로 즐거운 시각적 효과와 리듬감을 제공한다. 우리는 사용자의 컴퓨터 화면에서 음악에 맞춰 춤추는 인간형 댄스 모델의 모습을 실시간으로 시연할 수 있도록 하였다. 이는 춤을 좋아하는 사람들이 춤을 배우는 용도로 사용할 수 있고, 원하는 댄스 모델을 택하여 독특하고 무궁무진한 2차 창작 또한 할 수 있다.

심포니는 기존의 음악 플레이어와는 다른 파격적인 소리의 시각화를 갖추고 있다. 이러한 시각화는 재생 중인 소리를 분석하여 적절히 제공됨으로써 사용자는 청각과 시각을 동시에 사용하게 되고, 이는 사용자가 음악의 느낌을 보다 잘 공감할 수 있게 한다.

심포니는 자유로운 UI설정을 통해 노래 분위기에 최적화된 독특한 UX를 가지고 있다. 다양한 모드와 기능들을 통해 목적에 맞게 심포니의 형태를 변화시켜 사용할 수 있고, 이를 통해 시스템 자원 점유율 또한 적절하게 조절할 수 있다. 또한 더블클릭이나 스크롤 등의 조작 방식을 색다른 곳(테두리 등)에 적극적으로 도입함으로써 사용자에게 편리한 느낌을 제공하며, 소리의 시각화를 극대화시키는 동적인 디자인은 사용자의 창의력을 자극한다. 가사나 댄스, 플레이어 등의 세세한 작동 설정과, 사용자 스킨과 그 편집기는 사용자마다 가장 편안하고 자유로운 프로그램의 사용이 가능하도록 도와준다.

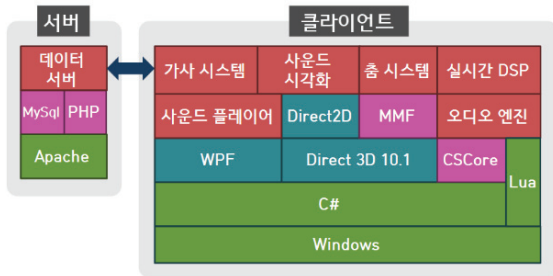
4. 기능 구현

4.1 심포니의 동작 흐름도



[그림 1] 동작 흐름도

4.2 심포니의 프로그램 구조도



[그림 2] 프로그램 구조도

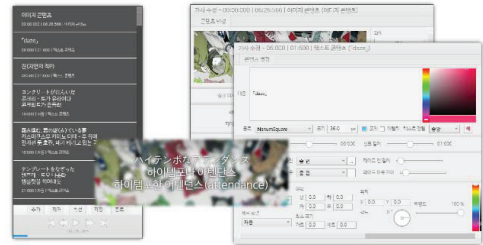
4.3 플레이어 인터페이스



[그림 3] 플레이어 UI

[그림 3]은 심포니의 스크린 샷이다. 심포니는 여러 가지 오디오 코덱 (MP3, WAV, FLAC, WMA, MP4 등) 을 지원한다. 사용자는 창 테두리를 더블 클릭하여 각 방향으로 창이 늘어나고 줄어드는 식으로 창 모드가 변경되어 사용하는 상황에 맞춰 변경할 수 있다. 단축키 F9키를 누르면 [그림 3]의 가장 좌측 하단에 보이는 미니플레이어를 팝업시킬 수 있다. 프로그램의 여러 가지 설정은 플레이어의 오른쪽 위에 보이는 회색 박스 안의 설정 버튼을 눌러 설정 창을 열어 변경할 수 있다. 플레이어 배경에는 오디오 비주얼라이저가 뜨게 되는데, 오실로 스코프, 스펙트럼 분석기, VU 미터를 표시할 수 있게 디자인 되어있다. 또한 사용자는 설정 화면을 통해 시각화와 관련된 설정을 변경할 수 있다.

4.4 가사 시스템



[그림 4] 가사 에디터

가사 에디터에서는 가사를 만들고, 저장하고, 업로드하여 사람들과 공유할 수 있는 기능이 구현되었다. [그림 4]은 가사 에디터로 가사를 편집하고 있는 것으로 현재 작업하는 가사의 리스트를 볼 수 있고, 제작된 가사를 실시간으로 재생하며 여러 가사 오브젝트를 동시에 수정할 수 있다. [그림 4]의 좌측에서 보이는 가사 오브젝트 에디터에서 가사의 페이드인/아웃 애니메이션, 레이아웃, 이펙트와 가사의 콘텐츠를 수정할 수 있다. 가사 콘텐츠 에디터는 가사 오브젝트가 출력할 콘텐츠를 수정하는 텍스트와 이미지의 두 종류의 에디터가 존재한다. 가사 시스템에서는 여러 가지 애니메이션 (페이드인/아웃, 블러, 회전, 날아오기 등), 애니메이션의 변속을 제어하는 KeySpline, 정렬이나 회전과 같은 레이아웃 옵션, 그림자와 블러 이펙트 등을 지원하여 사용자가 창의적으로 가사를 제작하고 애니메이션 할 수 있는 환경을 구현하였다.

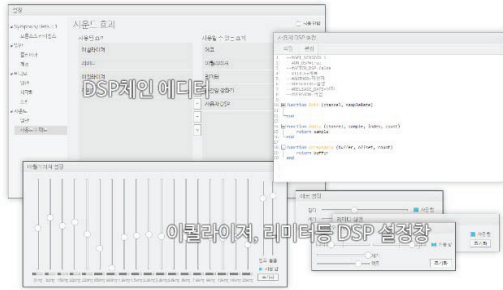
4.5 플레이어 스킨 시스템



[그림 5] 스킨 에디터

스킨 에디터는 설정>비디오>스킨에서 수정할 스킨을 더블클릭하여 사용할 수 있게 구현되었다. [그림 5]에서 볼 수 있듯이, 스킨에디터에서는 설정할 수 있는 스킨 리소스들의 리스트를 보여주고, 각 리소스에 맞는 에디터가 표시된다. 리소스의 종류는 이미지, 그래픽션, 단색 브러시가 있고, 플레이어에서 보이는 대부분의 브러시를 재정의 할 수 있게 구현되었다. 스킨의 편집 또한 실시간으로 플레이어에 적용된 결과를 볼 수 있다.

4.6 실시간 사운드 효과 (Digital Sound Processing; DSP)



[그림 6] DSP 설정 창

DSP 에디터에서는 플레이어의 사운드 이펙트를 실시간으로 편집할 수 있다. [그림 6]에서는 DSP들의 설정을 편집하고 있는 모습을 보고 있다. 기본 내장 이펙터(이퀄라이저, 리미터, 에코, 스테레오 인핸서)는 각각 내장 에디터를 가지고 있고, 사용자 DSP는 사용자가 직접 Lua를 통해 사운드 이펙트를 스크립팅 할 수도 있다. 간단하게 구현하고 싶은 노멀라이저와 같은 이펙터들을 실시간으로 구현해 넣을 수 있다.

4.7 춤



[그림 7] 춤을 보여주고 있는 화면

춤 에디터는 MMF 오픈소스 프로젝트를 기반으로 개발되었고, PMX와 VMD라는 MMD의 모델 포맷과 모션 포맷을 사용하며, 이는 Direct3D를 통해 WPF의 출력 컨트롤 안으로 렌더링 된다. 춤 불러오기 창에서 모델과 모션 파일을 불러온 뒤 연결된 곡을 재생하면 [그림 7] 과 같이 플레이어에서 실시간으로 렌더링 되는 모델의 춤을 볼 수 있다.

5. 결론 및 향후 계획

본 연구에서는 기존의 음악 플레이어가 가지고 있었던 정적인 분위기와 창작과 연결되기 힘든 환경을 개선하여, 음악적 공감을 돕고 그에 따른 창의적 창작 활동을 지원할 수 있도록 심포니를 개발하였다. 심포니는 음악적 공감을 돕기 위해 유연한 비주얼라이저와 프로그램 UI/UX를 가졌고, 창의적 창작 활동을 지원하기 위해 전문적인 기능까지 포괄할 수 있는 가상 시스템과 3D모델의 춤을 실시간으로 감상할 수 있는 춤 시스템을 구현하였다. 이로써 심포니는 음악 감상 목적으로도, 2차 창작의 목적으로도 활용될 수 있다. 이러한 환경을 다른 사용자와 같이 공감할 수 있고, 자신의 작품을 손쉽게 공유하고 퍼뜨릴 수 있도록 하여 사용자뿐만 아니라 많은 사용자로 하여금 음악적 공감을 느끼게 하고 창작의 기회를 줄 수 있도록 프로그램 커뮤니티를 구현하거나 모델과 모션 에디터, 음악 방송 구현 등으로 향후 발전해나갈 계획이다.

참고 문헌

- [1] ALSong. 2017년 07월 28일 검색 <http://www.altools.co.kr/Download/ALSong.aspx>
- [2] GomAudio. 2017년 07월 28일 검색 http://gom2.gomtv.com/release/gom_audio.htm
- [3] C#. 2017년 07년 28일 검색 <https://docs.microsoft.com/en-us/dotnet/csharp/csharp>
- [4] WPF. 2017년 07월 28일 검색 <https://msdn.microsoft.com/ko-kr/library/mt149842.asp>
- [5] Lua. 2017년 07월 28일 검색 <https://www.lua.org/>
- [6] 니시자와 유메지(2014). **My First MySQL & PHP Programming ~ 데이터베이스 기초부터 PHP 활용까지~** 부천: FREELEC.
- [7] 가와마타 아키라(2015). **C# 코딩의 기술 ~ 기본편, 똑똑하게 코딩하는 법 ~**, 서울: 길벗.
- [8] 가와마타 아키라(2016). **C# 코딩의 기술 ~ 실전편, 똑똑하게 코딩하는 법 ~**, 서울: 길벗.