

WESC-11-09-결과보고서

B L E N D

참가부문 : 스마트 챌린지 삼성전자

팀 명 : 陽 熙 知

팀구성원

No.	구분	성명	소속명	No.	구분	성명	소속명
1	팀장	곽지혜	경희대학교	6	팀원		
2	팀원	김남용	명지대학교	7	팀원		
3	팀원	김두현	한양대학교	8	팀원		
4	팀원			9	팀원		
5	팀원			10	팀원		

2011. 9. 28

1. 목차

2. 개요	3
2.1 작품명	3
2.2 작품 개요	3
2.3 목적	3
3.작품설명	5
3.1 Software 구성	5
3.2 Software 기능	6
3.3 프로그램 사용법	10
3.4 개발환경	11
4.프로그램 설명	11
4.1 파일구성	11
4.2 함수별 기능	12
4.3 주요 함수의 흐름도	13
5. 응용 분야	13
6. 기타	14
7. 제작자 정보	14
8. 개발 단계별 기간 및 투입 인원	14

2. 개요

2.1. 작품명

BLEND : (보기 좋게) 조합하다; '음원을 조합하다'라는 의미로 사용하였다.

2.2. 작품 개요

TV에서 가수들이나 개그맨들이 노래를 소재로 한 노래개그를 보여준다. 그 중에서도 비슷한 느낌의 노래를 두 개 또는 그 이상을 연결하여 마치 하나의 노래인 것처럼 자연스럽게 연결이 되는 곡들이 있다. 과거 유리상자부터 최근에는 박휘순까지 이러한 노래개그는 예전부터 지금까지 존재하고 있다.

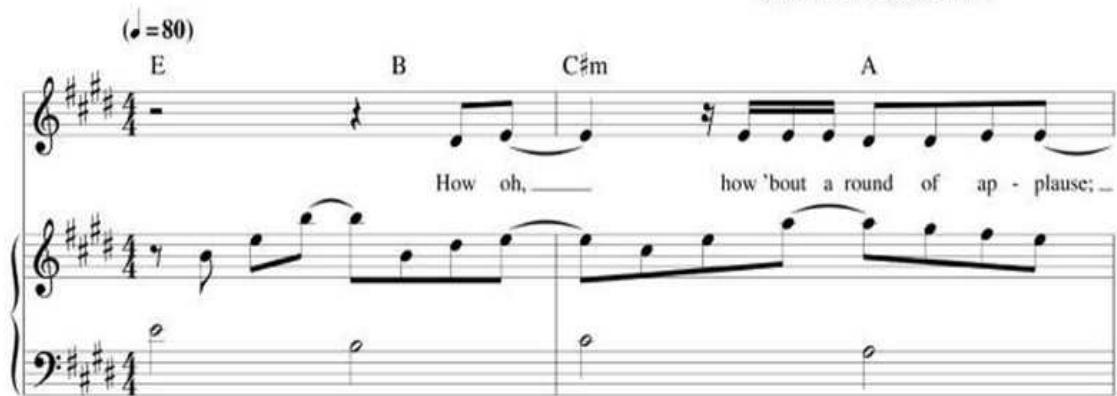
두 개의 노래를 연결하였는데도 이질감이 없는 이유는 두 노래의 화음이 비슷하게 진행되기 때문이다. 화음은 그 노래의 반주를 의미한다. 멜로디가 아니기 때문에 반주만 들어보면 비슷한 반주로 진행되는 노래를 많이 찾을 수 있다. 표절시비가 붙는 노래들은 대부분 **반주진행**이 비슷하기 때문이다.

그리고 세계적으로 히트한 노래들 중에서 마법의코드라 불리는 **E-B-C#m-A** 코드로 진행이 되는 곡이 많다. 모차르트의 'Canon'부터 시작된 이 코드의 진행은 비틀즈의 'Let It Be', 미카의 'Happy Ending', U2의 'With or Without You' 등등 이외에도 현재 알려진 많은 히트곡들이 코드로 진행 된다. 그렇기 때문에 이 네 개의 코드를 일명 머니코드(Money Chord)라고 불린다. 이 코드 이외에 I-V-VI-IV 로 진행되는 히트곡도 70곡 이상이 된다.

이러한 코드 이외에도 같은 코드로 진행이 되는 곡도 무수히 많다. 이러한 노래들 중에서 진행이 같은 노래를 분류하고 그 노래들을 연결시키면 또 다른 하나의 노래가 될 수 있다.

TAKE A BOW

Words and Music by
MIKKEL ERIKSEN, TOR ERIK HERMANSEN
and SHAFFER SMITH



일상에서 음악을 들을 때 몇 번 반복해 들으면 지루해지는 경향이 있다. 기존음악을 새로운 조합을 통해 사용자가 듣는 재미를 느끼게 하고자 한다. 이 프로젝트는 음원의 화음 진행을 분석하고, 진행이 유사한 두 개 이상의 노래를 이질감 없이 연결하여 새로운 음원을 만들어내는 어플리케이션이다.

2.3. 목적

상용화된 음악 재생 프로그램에서는 재생하는 방법은 순차재생방식과 랜덤재생방식이 있다. 이 외에도 iTunes의 Genius나 Sony의 Sense Me Channels처럼 같은 장르의 곡을 찾아 재생하는 방식이 있다. 우리는 이러한 방식 이외에 새로운 재생기법을 제안하여 사용자의 선택의 폭을 좀 더 넓히려고 한다.

음악에서 소리는 음이라는 단위로 표현할 수 있으며, 이 음들이 시간적 순서에 따라 배열되어 하나의 곡이 완성된다고 할 수 있다. 이들 음은 악보의 형식에서 음표로 기호화될 수 있으며, 지금까지 작곡된 수많은 음악들이 이렇게 이루어진 것이다. 따라서 어떤 음악을 컴퓨터가 처리할 수 있는 데이터로 변환하려면 음악 소리로부터 그 음을 추출하는 것이 하나의 방법이 될 수 있다. 다시말해, 특정 소리 음으로 부터 그 음을 나타내는 기호로 변환시켜 음표로 변환해야 한다. 음표를 이용해 높이, 길이, 강약등을 표시할 수 있는데, 이번프로젝트에서는 음표들을 순서대로 배열한 표현 형태인 멜로디를 이용한다. 멜로디를 추출하기 위해 다음과 같은 과정이 필요하다. 먼저 음악파일에서 반주 부분 제거를 통해 목소리 부분만 남기도록 필터링을 한다. 이렇게 얻어진 연속적인 신호를 주파수를 축으로 표현되는 스펙트럼(spectrum)의 형태로 변환시켜 입력신호의 주파수를 확인해야한다. 여기서는 FFT(FastFourierTransform) 방식을 사용한다. 얻어진 주파수로부터 우리가 인식하는 음계의 영역으로 표현하기 위해서는 평균율지식(EqualTemperament)을 이용하게 된다. 이 과정을 거쳐 변환된 멜로디 데이터는 컴퓨터가 처리할 수 있는 이산적이고 순차적인 특징을 가지게 된다. 생성된 멜로디를 가지고 코드를 찾게 되고 코드진행이 유사한 다른 음원을 찾는다.

이것을 조합하여 재생한다면 하나의 새로운 재생기법이 될 수 있다. 사용자의 편의를 고려해 일반적인 재생방법처럼 하나의 모드로 지원할 것이다.

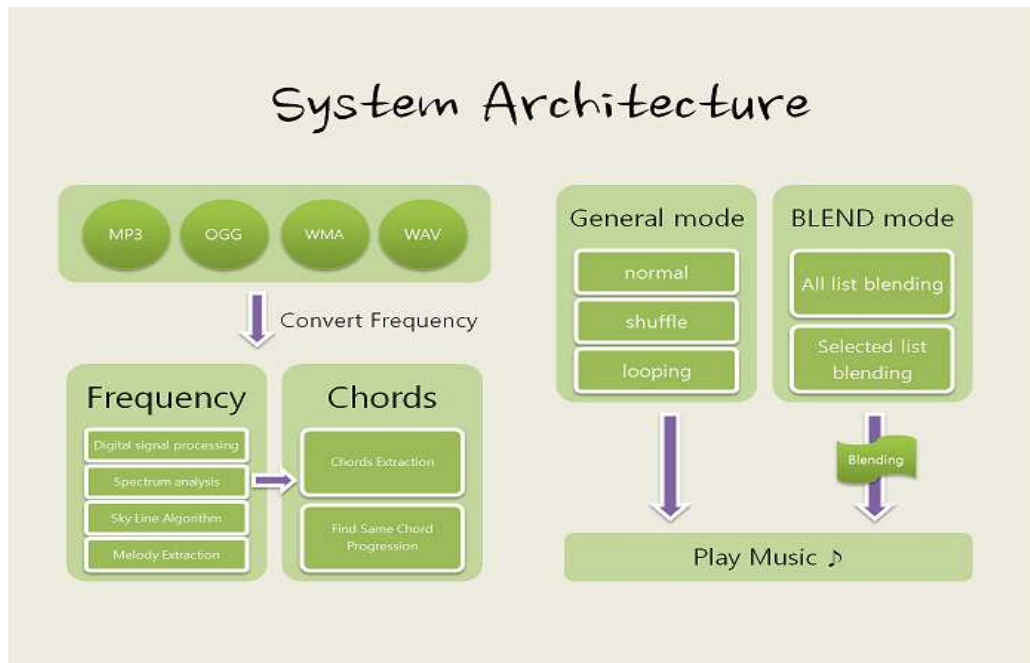
상용화된 음악재생 프로그램에서의 재생방법인 순차재생방식과 랜덤재생방식과는 다른 Blend라는 재생방식을 개발한다. Blend는 서로 다른곡에서 화음을 찾고 같은 순서로 화음이 진행되는 노래를 서로 연결시켜 두 곡 이상의 곡을 자연스럽게 연결시켜주는 재생방식이다.

이번 프로젝트에서 멜로디 추출방식은 음악파일을 wav 파일로 변환하여 얻어진 연속적인 신호를 주파수를 축으로 표현되는 스펙트럼의 형태로 변환시켜 입력신호의 주파수를 확인한다. 스펙트럼은 시간-주파수 분석에 중요한 도구로써 필터뱅크 구조와 움직이는 윈도우 FFT구조이다. 이렇게 얻어진 스펙트럼을 이용하여 최고점을 추출하는 알고리즘을 구현한다. 최고점을 찾는 함수는 주파수축을 따라 오직 1차원의 분석을 한다. 왜냐하면, 음악스펙트럼은 규정된 수평bias를 가지고 있다. 즉 음이 시간(수평)축을 따라 긴 주기에 있기 때문이다. 최고점을 위해 $B(k,l)$ 행렬의 각 행을 검색하여 음이 나타나는지 아닌지 그리고 그것이 얼마나 지속되는지(Duration)을 구할 수 있다.

또 디지털필터를 거치는 과정에서 필연적으로 연산량이 늘어나고 각 음계의 주파수와 지속시간을 저장하는 과정에서 시간이 걸리게 됐다. 그래서 멜로디추출을 한번 한 곡은 화음정보를 파일로 개별 저장하여 다음번에 그 곡은 멜로디 추출 과정에서 제외시킨다.

3. 작품 설명 (최대한 자세하게 기술)

3.1. Software 구성

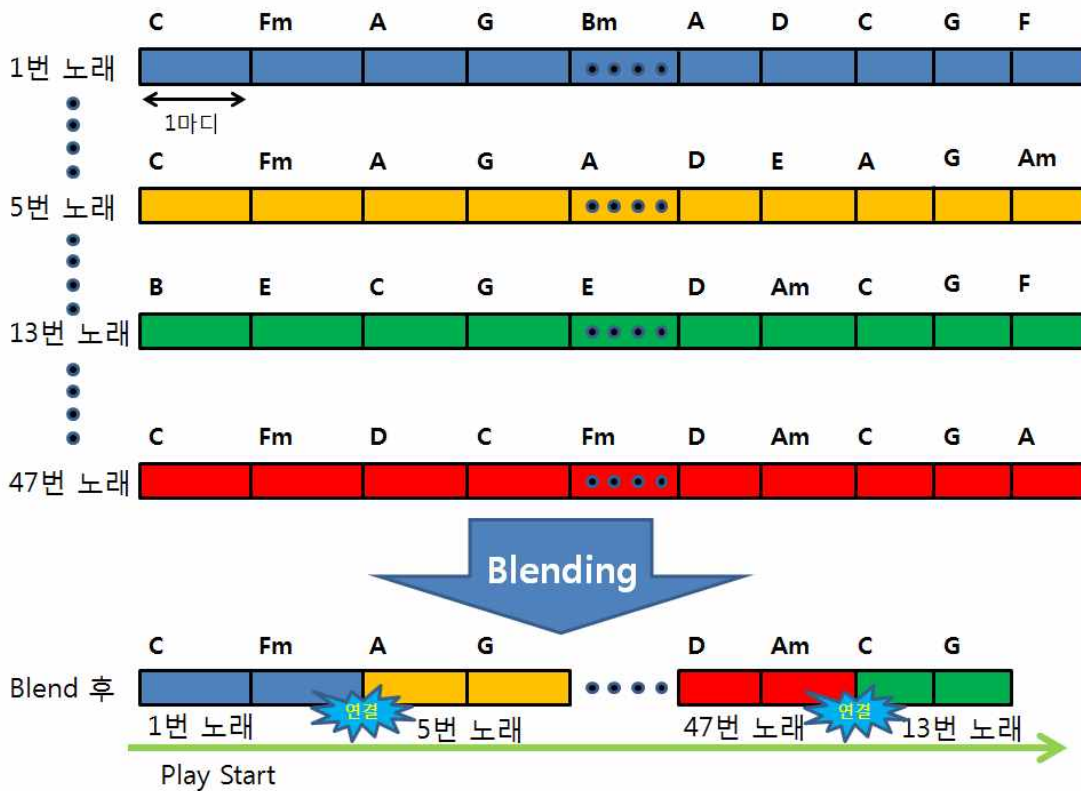


시스템의 전체 흐름도이다. 어플리케이션이 동작과 동시에 디바이스의 SDCard에 있는 음원파일을 찾는다. 찾은 음원파일은 MP3, WMA등을 지원하는데, 파일들은 변환과정을 거친다. 변환과정은 원곡에서 MR부분이 제거된 Vocal부분을 추출한다. 추출된 Vocal에서 스펙트럼 분석을 통해 멜로디라인을 추출하고 멜로디라인을 토대로 주파수매칭을 통한 음계추출을 한다.

이렇게 추출한 음계에 어울리는 코드를 마디별로 생성하여 음원의 전체적인 코드를 추출한다. 이것으로 코드진행이 유사한 다른 곡을 탐색한다.

'Blending' 기능을 통해 같은 코드로 진행하는 곡들을 플레이리스트에 나타내고 해당 구간을 재생한다.

3.2. Software 기능



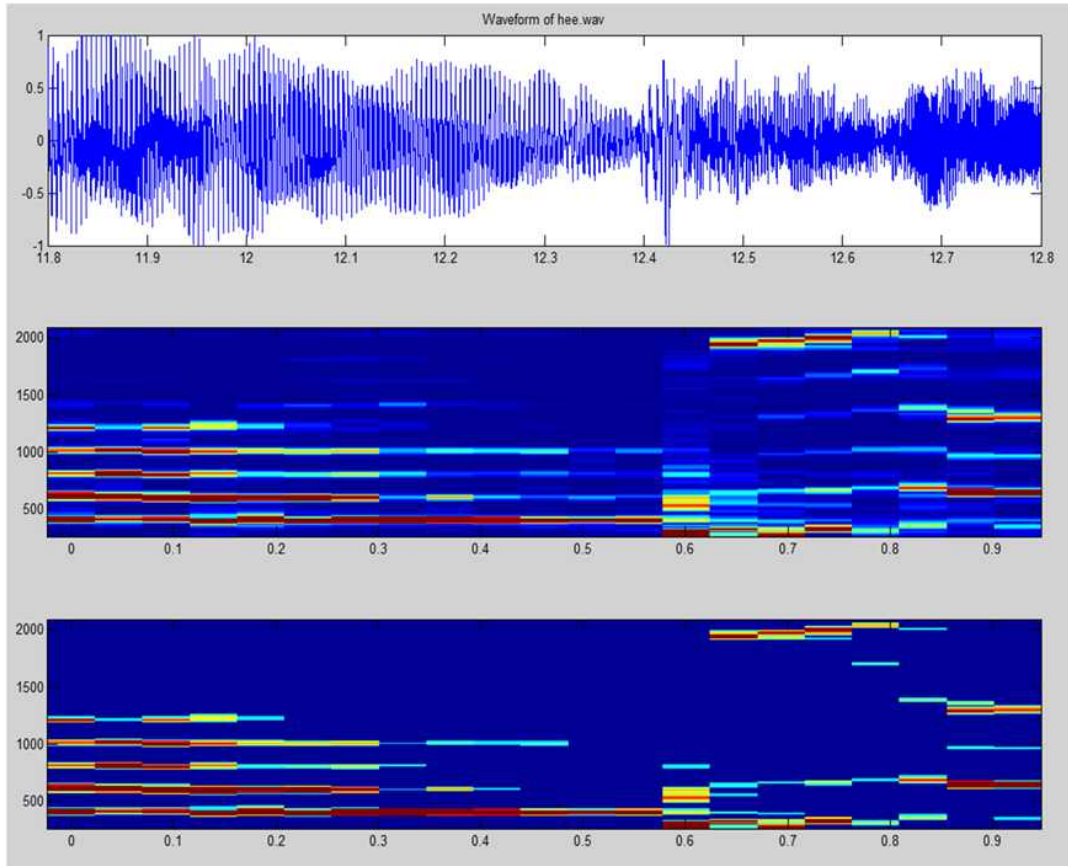
본 프로그램은 같은코드로 진행되는 노래를 찾아 서로 다른 노래를 연결하여 랜덤으로 재생 시키는 프로그램이다. 그렇게 하기위해 먼저 음악파일을 분석하여 각 마디를 구분하고 마디별 음계를 추출 후 그 음계를 토대로 화음을 추출하는 것이 1차 목표이다. 그 후에 같은 코드 진행이 되는 음악파일을 찾고 이를 연결 해주는 것이 다음 목표가 되겠다. 마지막으로 연결이 된 음악을 재생하는 것이 마지막목표이다. 연결방식은 위 그림과같이 A라는 곡이 처음부터 재생이 되다가 B곡과 같은 코드인 부분이 나오면 자연스럽게 B곡으로 넘어가 재생을 한다. 마찬가지로 B곡이 재생하다가 B곡과 C곡에서 같은 코드부분이 나오면 자연스레 C곡으로 넘어가는 방식으로 재생 한다.

1) MP3신호처리

가장 보편적인 음원으로 사용되는 MP3파일에서 코드를 분석하는 것이 목적이다. 하지만 MP3 이외의 다른 포맷의 음악 파일도 코드를 분석하기 위해서는 시간영역, 주파수영역에서 분석을 하는 2가지 방법이 존재한다. 하지만 시간영역으로 분석 하는 것은 거의 불가능한 방법이다. 따라서 시간영역의 신호를 주파수 영역으로 바꾸는 신호처리 알고리즘을 사용함으로써 음원에서 코드분석을 가능하게 한다. 이 때 사용되는 방법으로는 푸리에 변환공식을 사용하여 처리를 해준다. 그리고 단순히 푸리에 공식을 사용하면 처리속도가 매우 저하되므로 FFT(FastFourierTransform)를 사용하여 처리 한다.

2) VocalExtraction

MP3파일상에는 보컬, 기타, 베이스, 피아노, 드럼등 여러악기의 음원이 합쳐져 있다. 그러나 각각의 악기는 주파수 대역폭이달르고, 음역에서 차이가있어 필터링을 통해 멜로디를 뽑아낼 수 있다. 하지만 완전한 분리는 불가능 하기 때문에 필터로 걸러진 주파수 대역에서 FFT를 통해 계수값이 가장 큰 값을 뽑아 멜로디라인을 형성할 수 있다. 사람이 낼 수 있는 음역은200Hz(G3)~2,093Hz(C7)이다. 따라서 음원에서 200Hz~2,093Hz대역을 뽑으면 사람의 목소리에 대한 주파수를 얻을 수 있다. MP3파일에서 보컬부분을 뽑은 후 스펙트럼 분석결과와 아래의 그림과 같다.



첫 번째 그래프는 실제 시간 영역에서 본 음원의 파형 모습이고, 두 번째 그래프는 FFT변환을 통해 주파수 값을 시간의 변화에 따라 출력한 결과 이다. 마지막 그래프는 임계값(Threshold)을 정하여 그 이상의 값은 나타내지 않게 출력했다. 따라서 이 그래프의 Y축값에 해당하는 주파수의 음정을 찾아 멜로디 라인을 구성할 수 있다.

3) 멜로디와 화음 관계 분석

작곡을 할 경우 멜로디를 만들어 놓고 코드를 정하는 방법과 코드를 정해 놓은 다음에 멜로디를 작성하는 방법이 있다. 이 작품에서는 멜로디 라인에서 코드를 추출과정을 필요로 한다. 그 이론을 설명하면 다음과 같다. 화음에는 장조3화음, 단조3화음이 존재한다. 장조3화음의 경우 1도(C=CM), 4도(F=FM), 5도(G=GM)화음이 있고 이는 각각 <도,미,솔>, <파,라,도>, <솔,시,레>의 음을 갖고 있다. 단조 3화음의 경우에는 1도(Am=a), 4도(Dm=d), 5도(Em=e)화음이 있고, 이는 각각 <라,도,미>, <레,파,라>, <미,솔#,시>의 음을 갖는다. 장3화음은 두 개의 음정사이에 반음이 존재하지 않는 것을 말하고, 단3화음은 두 개의 음정사이에 반음이 1개 존재하는 것을 말한다. 장조3화음의 경우는 장3화음이 밑에 오고, 단3화음이 위에온다. 반대로 단조3화음의 경우는 단3화음이 밑에오고, 장3화음이 위에 온다. 멜로디 라인에서 코드를 찾는 방법은 다음과 같다. PassingTone (한음에서 다음 음으로 진행하기 위한 음으로 코드 구성과 상관 없는 예외상황의 음)을 제외하고 멜로디의 순서를 지키면서 여러 코드중에서 적합한 한 가지 경우를 선택한다. 예를들면, 멜로디의 진행이 <라,솔,도>라면 각각의 코드의 음들을 찾아 본다. 이때, <라,솔,도>가 포함된 코드는 존재하지 않고 <라,솔>, <솔,도>, <도,라>가 포함되는 코드를 찾아보면 C코드가 <솔,도>를 포함하고 있다. 진행도 <도,미,솔>로 <솔,도>의 진행을 지키고 있으므로 이 멜로디의 진행코드는 C코드로 결정하게 된다.

4) 멜로디추출(스카이라인)

음악에서 멜로디를 추출하는 것은 연속된 시간상에서 입력된 여러음들 중에서 하나의 대표음을 선택하거나, 대표음이 없는 경우 아무것도 선택하지 않는 과정의 연속으로 볼 수 있다. 음을 선택하는

과정에서는 특정한 기준을 사용하는데, SkylineAlgorithm의 예에서도 알수 있듯이 어느 상황에서나 적용 가능하며 음악적으로 의미 있는 원칙을 찾기는 매우 어렵다. 예를들어 아래 그림과 같은 음악의 경우 Skyline Algorithm을 그대로 적용해도 멜로디를 추출하는데 특별한 문제가 없다.



<쇼팽의 군대 폴로네이즈>

동시에 발생하는 음들이 모두 같은 지속시간을 갖고 있으므로 어떤 음도 임의로 지속시간을 줄일 필요가 없고 항상 가장 높은 음이 멜로디이기 때문이다. 그런데 아래와 같은 음악에서는 하단의 오선지에 표시된 음들이 멜로디가 되고 상단의 오선지에 표시된 음들이 반주가 된다.



<베토벤의 월광 3악장>

그러므로 동시에 발생하는 음들 중에서 높은 음이 멜로디에 속한다는 원칙은 이 상황에서는 적용 가능하지 않다. 그리고 하단의 멜로디가 진행되는 도중에 상단의 반주들이 진행되고 있기 때문에 이 경우 Skyline Algorithm을 그대로 적용한다면 아래 그림에서 보이는 바와 같이 상단의 반주들이 멜로디에 포함되게 된다. 따라서 위의 <베토벤의 월광 3악장>과 같은 경우에는 Skyline Algorithm의 적용이 어렵다.



<베토벤의 월광 3 악장에서 Skyline Algorithm 을 이용해서 멜로디를 추출한 결과>

그러므로 음악에서 멜로디를 추출하기 위해서는 각 음악의 유형에 따라 적합한 원칙을 사용하는 것이 정확한 멜로디를 추출 하는데 필요하다. 예를들어 <쇼팽의 군대 폴로네이즈>의 음악과 같은 경우 다음과 같은 원칙을 이용해서 멜로디를 분류할 수 있다.

첫째, 동시에 발생하는 음의 경우 보다 낮은음이 멜로디에 속한다.

둘째, 선택된 낮은음이 지속되는 기간 동안 입력되는 음은 모두 반주이므로 멜로디에 포함시키지 않는다.

위의 두 가지 원칙을 이용하면 <쇼팽의 군대 폴로네이즈>와 같은음악에서 저음부의 멜로디를 추출할 수 있다.

또 하나의 예를들면 <베토벤의 월광 3악장>과 같은음악은 다음과 같은 원칙을 이용해서 멜로디를 분류할 수 있다.

첫째, 동시에 발생하는 음들 중에서 가장 긴 지속시간을 갖는 음이 멜로디에 속한다.

둘째, 동시에 발생하는 음들 중에서 두 음 이상이 같은 길이의 가장 긴 지속시간을 갖는다면 그 중에서 가장 고음이 멜로디에 속한다.

셋째, 멜로디에 속하는 음이 지속되는 시간동안 입력되는 음들은 모두 반주이므로 멜로디에 포함시키지 않는다.

이처럼 음악의 유형에 맞춰 적절한 원칙을 사용하여 멜로디를 추출 한다면, 음악의 구조적 특성을 고려하지 않고 동일한 원칙만을 사용하는 기존의 알고리즘들에 비해 보다 정확한 결과를 얻을 수 있었다.

5) 마디구분(Beats Per Minute, BPM)

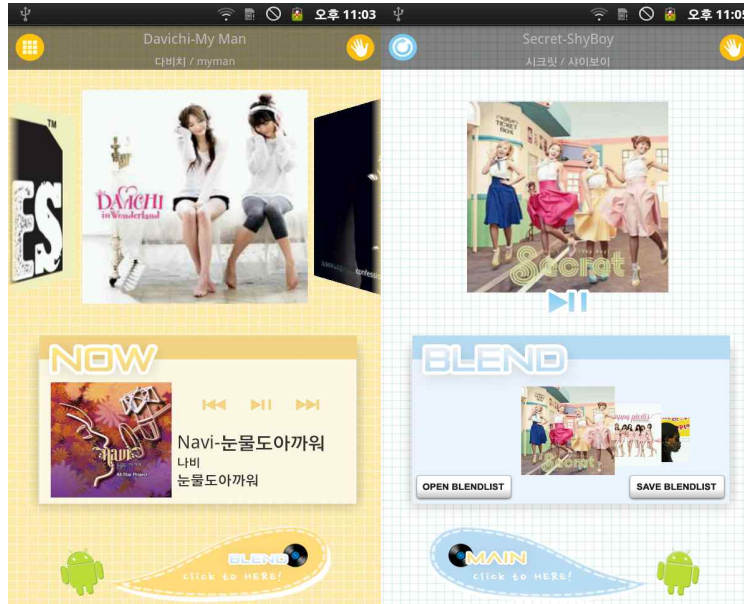
마디는 악보에서 세로줄로 구분되어 있는 악보의 가장 작은 단위로 곡 머리의 박자표에 의해 한 마디에 들어가는 음표나 쉼표가 결정된다. 예를들어, 4분의 4박자는 1마디 안의 음표 및 쉼표의 합계가 4분음표 4개에 해당 한다는 것을 뜻한다. 예외적으로 여린박으로 시작하는 노래의 경우 첫마디가 규정된 박수를 갖지 못 한마디(못갖춘마디)가 되고, 끝마디를 합해서 완전한 박수(갓춘마디)가 된다.

화성에서 마디를 분할하는 이유는 마디안에 있는 멜로디 구조가 하나 이상의 화음을 요구하기 때문이다. 즉 화음을 구하기 위해서는 마디를 구분 해야 한다. 마디를 나누는 방법은 우선 음원의 BPM(BeatPerMinute, 분당비트수)를 구한다. BPM은 1분에 몇 번의 비트가 발생 하는지를 표시 하는 것으로 현재 가요의 대부분이 4분의 4박자로 나오기 때문에 4분 음표하나를 한 박자로하여 1분에 몇 개의 4분음표가 나타나는지를 보여 준다. 그래서 60BPM은 1분에 60개의 4분음표가 있고 1초에 한 번 씩 소리가 나게 된다. BPM이 138인 노래라고 했을 때, 1분 내에 138개의 Beat가 존재하고 4Beats가 1마디가 된다.

한 마디가 걸리는 시간은 1분당 존재하는 마디의 개수 = $(BPM/4)$ 이므로 $34.5=138/4$. 즉, 138bpm의 노래의 경우 분당 34.5개의 마디가 존재한다. $60초/34.5=1.739초$ 이고 이는 1마디당 걸리는 시간이 된다. 멜로디 추출을 할 때 구한 Duration에 음표와 쉼표의 지속시간이 저장되어 있으므로 1.739초 마다 들어가는 음표와 쉼표를 구할 수 있다.

이 한마디를 분석하여 화음을 구하게 되는 것 이다. 이러한 일련의 과정을 하기 위해 BPM을 노래에서 추출 하는 것이 필요 하였다.

3.3. 프로그램 사용법 (Interface)

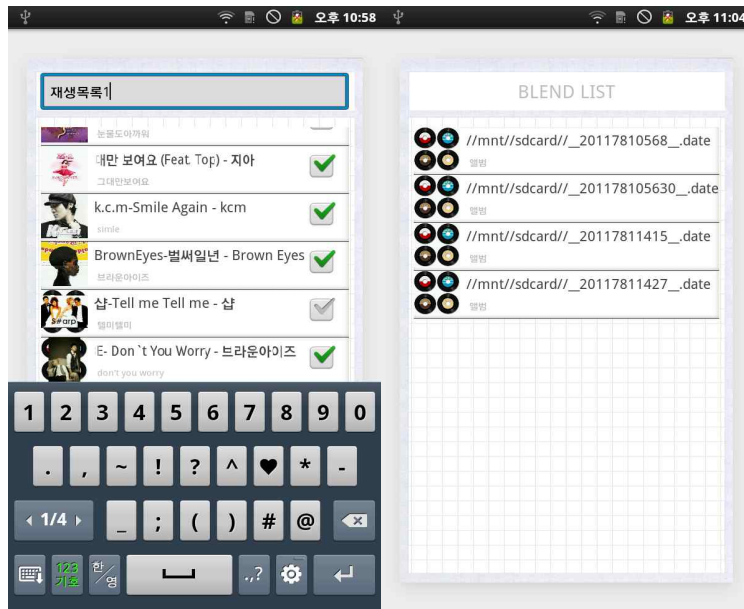


메인 화면의 모습이다. 원하는 음악을 재생할 수 있고 커버플로우를 통해 음악을 선택할 수 있다. 중간부분의 박스에서 현재 재생중인 음악의 정보를 알 수 있고, 일반재생을 할 수 있다.

Blend탭을 누름과 동시에 코드진행이 유사한 곡들이 랜덤으로 조합되어 리스트에 보여지고, 음악을 재생한다. 만약 리스트에 있는 조합을 새로이 하고 싶다면 Reblending를 클릭한다.



재생 목록을 불러와 원하는 재생 목록으로 음악을 들을 수 있고 재생 목록 추가기능을 이용하여 원하는 곡을 재생 목록에 담을 수 있다. 그리고 삭제, 저장버튼을 이용하여 원하거나 원하지 않는 재생목록을 삭제할 수 있다.



블랜드 재생 목록을 불러와 원하는 음악을 들을 수 있고 블랜드 재생 목록 추가 기능을 이용하여 원하는 곡을 재생 목록에 담을 수 있다.

3.4. 개발환경 (언어, Tool, 사용시스템 등)

- WindowsXP,Windows7
- Matlab2010
- VisualStudio2010
- Eclipse(AndroidSDK2.2ver)
- MultimediaLibrary

4. 프로그램 설명 (최대한 자세하게 기술)

4.1. 파일 구성

설치파일은 Blend.apk로 배포한다.

Code

- BlendActivity.java
- Blending.java
- CoverFlow.java
- dataillList.java
- FileManager.java
- java_comvertingclass.java
- java_convertingclassRemote.java
- java_convertingMCRFactory.java
- Madeby.java
- MusicUtils.java

- PlayActivity.java
- PlayList.java
- PlayService.java
- BlendInterface..aidl
- IRemoteServiceCallback.aidl

Manifest

- AndroidManifest.xml

Resource

- drawable,layout

4.2. 함수별 기능

함수명 : playBlendSongTrans(MediaPlayer, Music, double, double,
MediaPlayer, Music, double, double, boolean)

설 명 : blend 페이지에서 blend list를 재생시키는 함수. 다만 Music 형의 정보에서 파일의 path정보만 string으로 넘겨주기 위해 거치는 중간단계 함수이다.

함수명 : playBlendSong(MediaPlayer, String, double, double,
MediaPlayer, String, double, double, boolean)

설 명 : playBlendSongTrans에서 호출 되는 함수로써 블랜드 알고리즘을 구현한 핵심함수이다. 음악 파일에 대해서 시간정보와 루프 정보를 가지고 블랜드 리스트의 순서대로 음원을 재생시킨다. 이전 음원이 끝나가는 시점에 페이드인, 페이드아웃을 적용시켜 다음 음원을 정해진 마디시간 정보와 구간정보를 갖고 시작점을 찾아 재생시킨다. 두개의 MediaPlayer객체를 컨트롤 하여 음원들이 겹쳐지는 부분에서도 일시정지 및 종료가 가능하다.

함수명 : mwmcrWrite(String, String, int)

설 명 : sdcard내에 존재하는 음원 파일들을 마디별로 음을 분석해서 파일로 생성해주는 함수. 음원파일이 전부 지워졌다가 일부만 바뀐 음원을 넣었을 때 모두 분석을 하지 않도록 매칭 알고리즘이 짜여져 있다.

함수명 : mwmcrAnalysisCode(String, String, int)

설 명 : 음이 분석된 파일을 가지고 코드를 찾아내어 코드 값들을 마디별로 파일에 쓰는 함수.

함수명 : mwmcrStartTime(String, String, int, int)

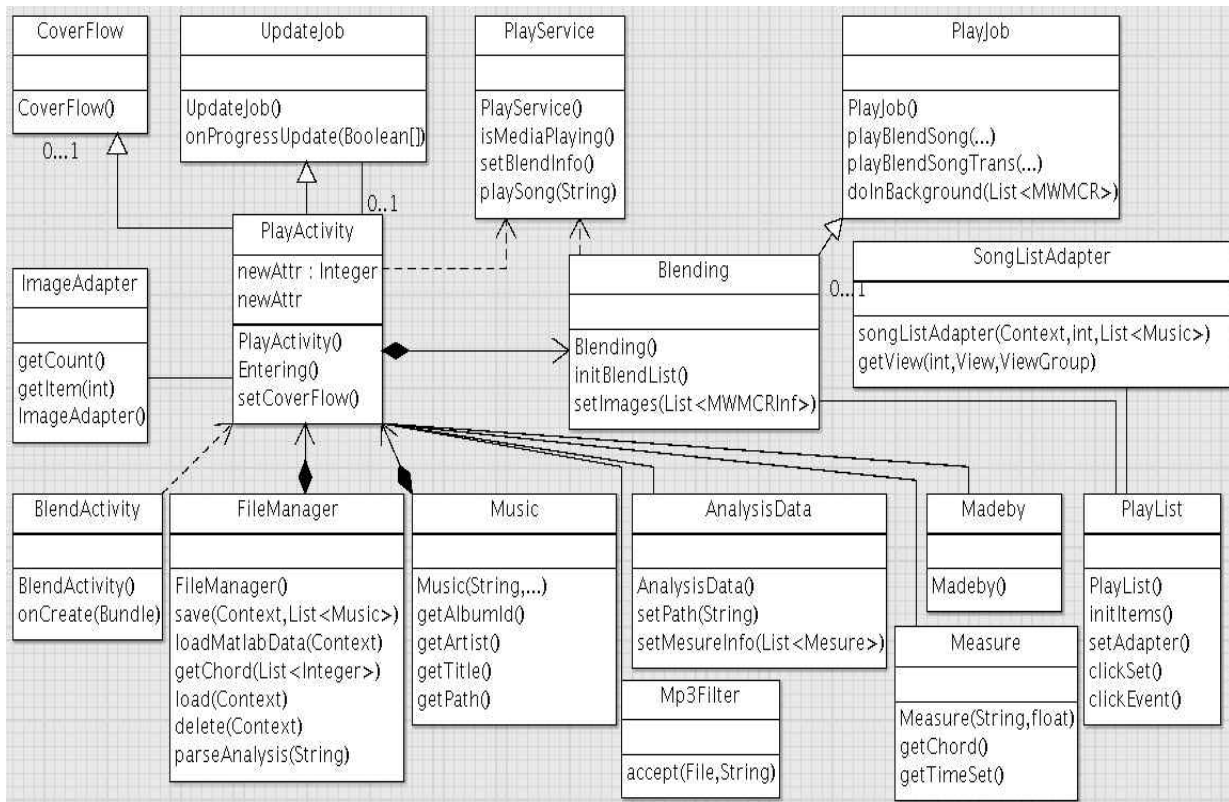
설 명 : 코드별로 분석 된 음원에서 블랜드로 이어질때 필요한 시작 시점과 루프 구간 정보를 파일에 쓰는 함수.

//ui 함수 TM기

함수명 : PlayInfo(String, String)

설 명 : 블랜드 리스트 목록을 만들어 List형식으로 만들어 넣어주는 함수.

4.3. 주요 함수의 흐름도



4.4. 기술적 차별성

5. 응용 분야

스마트폰 유저가 점점 늘어나고 있는 추세에서 많은 Applicaton이 개발 되고 있지만 음악을 듣는 Application은 기본으로 제공되어지는 기본 Application을 주로 사용하고 있다. 그러한 추세에 좀더 직관적인 UI를 가진 음악 플레이어를 제공하고 또한 거기에 Blend라는 기능을 추가시켜 사용자들에게 듣는 재미를 제공한다.

Blend는 기존의 음악플레이어의 주요 재생 방법인 순차재생과 랜덤재생 이외에 곡 중에서 같은 진행의 노래를 찾아서 사용자에게 제공하는 새로운 플레이 방식이다. 기존의 컴퓨터 음악플레이어나 MP3 플레이어에도 이 기능을 제공하면 스마트폰 유저가 아니라 컴퓨터나 MP3를 주로 사용하는 사람들에게도 듣는 재미를 줄 수 있다.

또한 이 프로그램에서 사용된 알고리즘을 적용시키면 음악의 코드를 뽑아내주는 프로그램이나 악보를 생성해주는 프로그램 또한 개발이 가능하다. 그리고 사람의 허밍을 녹음하여 위의 알고리즘을 적용하면 작곡을 해주는 프로그램도 만들 수 있다.



<악보생성프로그램 예>

6. 기타 (해당되는 경우에만 작성)

7. 제작자 정보

No.	구분	성명	소속(학교)	부서(학과)	직위(학년)	담당업무
1	팀장	곽지혜	경희대학교	전자전파공학과	3학년	PL, Design & UI
2	팀원	김남용	명지대학교	전자공학과	4학년	음원분석
3	팀원	김두현	한양대학교	전자통신학과	4학년	알고리즘개발

8. 개발 단계별 기간 및 투입 인원

이름	역할	1 주	2 주	3 주	4 주	5 주	6 주	7 주	8 주
김남용	BPM 분석 및 마디구분								
	디지털 필터 설계								
	음악파일 포맷 변경								
김두현	Blend 알고리즘 추가								
	원하는 곡만 블랜드 알고리즘 적용								
	추가되는 부분 Converting 작업								
곽지혜	기본재생목록과 블랜드재생목록 저장 알고리즘								
	Application 디자인 & UI 작업								
	위젯 구성								
공통	통합과정, 디버깅작업								