平成 25 年度　卒業論文

音の可視化を取り入れた Android アプリ

函館工業高等専門学校　情報工学科 5 年
東海林研究室　金澤しほり・清水愛未

目次

1 章　序論

1.1　研究背景

近年、スマートフォンやタブレットを用いた障害者向けのサービスやアプリケーションが普及し始めている。例えば、入力された文章を端末が読みあげて、健常者とのコミュニケーションを支援する聴覚障害者向けのアプリケーションなどがあげられる[1] 。

一方、音楽は今や娯楽として我々に必要不可欠になってきているが、聴覚障害者は健聴者と同様に音楽を聴いて楽しむことが困難である。しかし聴覚障害者は目や手で音楽を感じることは可能である。

そこで我々は聴覚障害者が MIDI キーボードによる演奏を通じて音楽を楽しむことができる Android アプリケーションを開発した。聴覚障害者が普通に MIDI キーボードを演奏しても音を聞き取ることが困難であるため不服や退屈を覚えてしまう。一方、我々が開発したアプリケーションを用いることで自分の演奏に合わせて画面上に表示されているキャラクターを動かすことができるので、聴覚障害者も視覚的に音楽演奏を楽しむことが可能である。

1.2　研究目的

本研究は聴覚障害者が音楽演奏を楽しむことが出来る Android アプリケーションの開発を目的とする。さらに開発したアプリケーションのテストをおこない聴覚障害者が音楽演奏を楽しむことが出来るか調査する。

1.3　英文アブストラクト

It is difficult to listen to music for hearing-impaired people because their ears cannot hear a sound. However, they can enjoy music through vibration handed down to their bodies. Recently, there are various techniques and applications that convert frequency of music into vibration to feel music for hearing-impaired people. Therefore we developed an android application with which hearing-impaired people can play a keyboard and enjoy music visually. When a user connects a MIDI keyboard to the android terminal and plays music, the screen of the terminal displays the animation of a character dancing with the music. In this study I implemented the movement of the character corresponding to the sound. Finally we tested the application under the same condition as hearing-impaired people.

2 章　アプリケーション概要

2.1　仕様

初めに Android 端末と MIDI キーボードを USB ケーブルで接続する。ユーザーがキーボードを叩き演奏を始めると、入力された音楽に合わせて画面に表示されているキャラクターが動く。そのアニメーションを通じて聴覚障害者でも音楽演奏を視覚的に楽しむことができる。

2.2　画面構成

開発するアプリケーションのタイトルは「DROID DANCE」である。このアプリケーションはタイトル画面、メイン画面、説明画面の三つの画面で構成されている。

2.2.1　タイトル画面（図 2.2.1）

アプリケーションを起動したときに表示される画面である。画面上のスタートボタンを押すとメイン画面に移動する。使い方ボタンを押すと説明画面に移動する。

図 2.2.1　タイトル画面

2.2.2　メイン画面（図 2.2.2）

MIDI キーボードを接続して演奏することで、この画面の中央に表示されているキャラクター「ドロイド君」が音楽に合わせて動くアニメーションが再生される。画面下部にある終了ボタンを押すとアプリケーションが終了する。

図 2.2.2　メイン画面

2.2.3　使い方画面（図 2.2.3）

アプリケーションの使い方を説明している画面である。画面下部にある戻るボタンを押すとタイトル画面に戻る。

図 2.2.3　説明画面

3 章　開発環境

開発環境は以下のとおりである。

CPU　　：IntelR Core™2 Duo CPU E7400 @ 2.80GHz × 2
OS　　：Linux Ubuntu12.04 LTS
Android 端末　：Nexus7(2012)
MIDI キーボード：KORG nano KEY
開発言語　　：Java
使用ソフト　　：Eclipse , Android Developer Tools ver22.2.1 , GIMP ver2.6.12
使用ライブラリ　：MIDI Driver

4 章　演奏とアニメーション

4.1　使用キャラクター

今回開発するのは Android アプリケーションであるので、スマートフォンや PC タブレット用の Android マスコットキャラクターである「ドロイド君」を使用することにした（図 4.1）。

図 4.1　ドロイド君

4.2　アニメーション

動作がなるべく滑らかになるように1種類のアニメーションにつき3～5枚程度の画像をつなげて使用するフレームアニメーションという手法でおこなった[2]。1オクターブ分の音階のキーボードに一つずつアニメーションを割り当て、全部で12通りのアニメーションを作成した。動きの種類は、ジャンプ（図 4.2-1）、回転（図 4.2-2）、バンザイ（図 4.2-3）、手をふる（図 4.2-4）、パタパタ（図 4.2-5）、ウィンク（図 4.2-6）、泣く（図 4.2-7）、きょとん（図 4.2-8）、怒る（図 4.2-9）、喜び（図 4.2-10）、幸せ（図 4.2-11）、悲しむ（図 4.2-12）である。感情を表す動きは、それが伝わるように作成した[3]。これらの各画像を xml ファイルにアニメーションとして記述した。

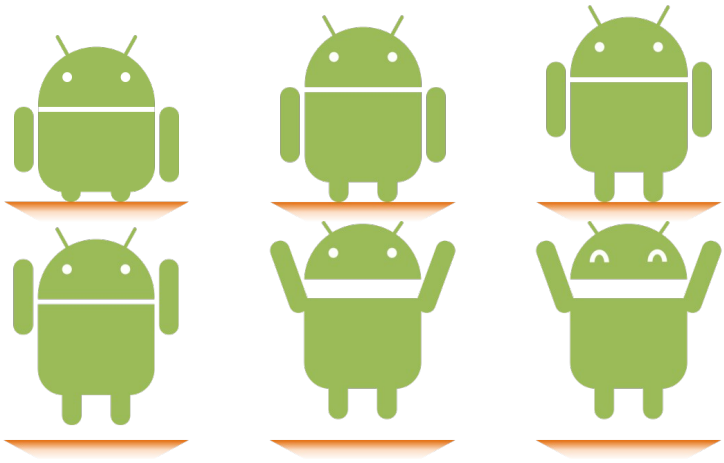

図 4.2-1　ジャンプ（droid_jump.xml）

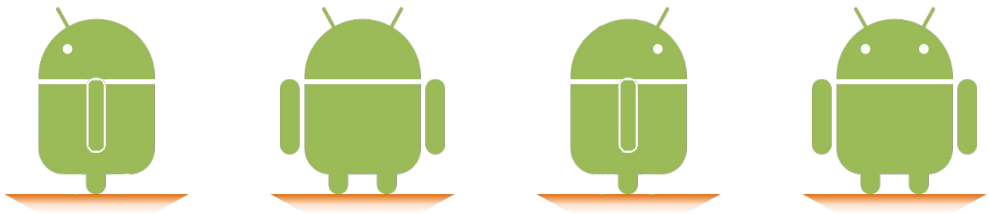

図 4.2-2　回転（droid_kaiten.xml）

図 4.2-3　バンザイ(droid_nicoban.xml)

図 4.2-4　手を降る(droid_furifuri.xml)

図 4.2-5　パタパタ(droid_patapata.xml)

図 4.2-6　ウィンク(droid_wink.xml)

図 4.2-7　泣く(droid_cry.xml)

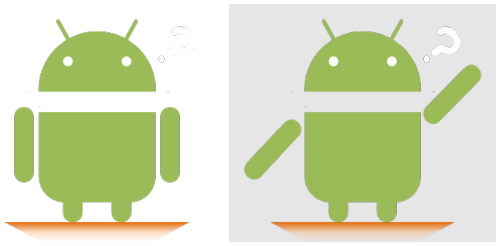

図 4.2-8　きょとん（droid_kyoton.xml）

図 4.2-9　怒る（droid_angry.xml）

図 4.2-10　喜び（droid_happy.xml）

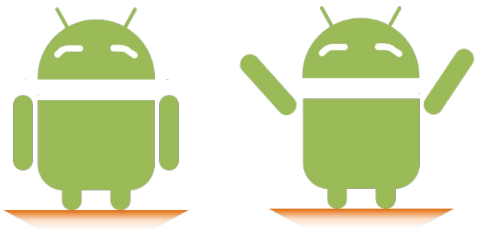

図 4.2-11　幸せ（droid_siawase.xml）

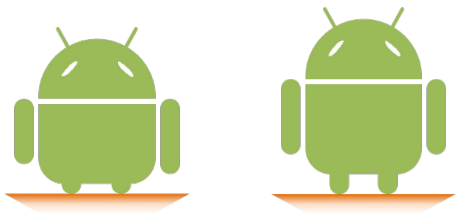

図 4.2-12　悲しみ（droid_sad.xml）

4.3　MIDI Driver

Android 端末と MIDI キーボードを接続するために Android USB MIDI Driver を使用した[4]。Android USB MIDI Driver とは MIDI を使うためのライブラリで Android3.1 以降のタブレットやスマートフォンに MIDI 対応の楽器や機器を繋げることができるようになる。

このライブラリを用いて我々はキーボードを演奏した時に画面に表示されているキャラクターをリアルタイムに動かしている。具体的には Android 端末を MIDI メッセージの受信側とし、MIDI キーボードからユーザーが入力した音を取得してキャラクターが音楽に合わせて踊っているようなアニメーションを表現した（図 4.3）。

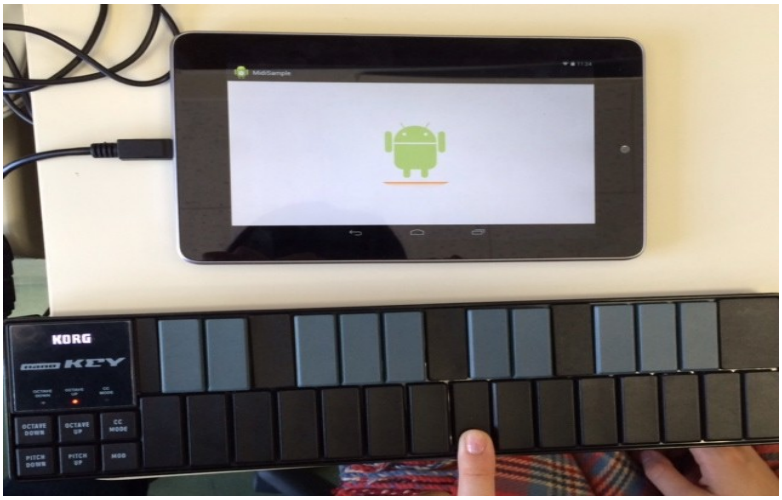

図 4.3　実際に演奏している様子

4.4　onMidiNoteOn と onMidiNoteOff

MIDI Driver には onMidiNoteOn と onMidiNoteOff という 2 つのメソッドが存在する。OnMidiNoteOn は特定チャンネルの特定の音（ノートナンバー[5]）が MIDI キーボードで押された時に呼び出されるメソッドである。また onMidiNoteOff は MIDI キーボードから指を離した時に呼び出されるメソッドである。主にこの 2 つのメソッドを利用してアニメーションを実装した。なおどちらのメソッドも引数からノートナンバーを取得できる。

4.5　ノートナンバー

ノートナンバー[5]とは、MIDI キーボードの音程を表すための数字である。ピアノ鍵盤の中央のCを 60 として、そこから半音ずつ順に低音は0まで、高音は 127 までの値が各鍵盤に割り振られている。例として、下の図 4.5 に開発で使用した MIDI キーボード上に対応するノートナンバーを示した。

図 4.5　ノートナンバー

4.6　アニメーションの実装

4.6.1　onMidiNoteOn での処理

ユーザーが MIDI キーボードのある鍵盤を押したときにメソッド onMidiNoteOn が呼び出される。図 4.6.1 に onMidiNoteOn の処理を示す。ここで引数 note には押された鍵盤のノートナンバーが入っている。

① 論理型配列 noteNumber[note]に true をセットする。この配列 noteNumber[note]は全ての鍵盤に対する現在の状態を記録しておく配列である。0～127 までノートナンバーがあるので配列サイズは 128 とする。true が鍵盤が押されている状態を示し、false は鍵盤から指が離れている状態を示す。

② ノートナンバー note を 12（音階）で割った余りに対応するアニメーションをアニメーション配列 noteAnime から取り出してアニメーションビューにセットする。

③ ビューからアニメーションを取り出す。

④ アニメーションを開始する。

```
public void onMidiNoteOn(final MidiInputDevice sender, int cable, int channel, int note, int
velocity) {
        ～略～
        noteNumber.set(note,true);                          //① 押された鍵盤に対して true をセット
        mImageAnimation.setBackgroundResource
        ( noteAnime[ note % 12 ] );                         //② アニメーションをセット
        AnimationDrawable anim =  (AnimationDrawable)
        mImageAnimation.getBackground();                    //③ ビューからアニメーションを取り出し
        anim.start();                                       //④ アニメーション開始
}
```

図 4.6.1　onMidiNoteOn の処理内容

4.6.2　onMidiNoteOff での処理

ユーザーが MIDI キーボードのある鍵盤から指を離したときに onMidiNoteOff が呼び出される。図 4.6.2 に onMidiNoteOff の処理を示す。やはり引数 note には押された鍵盤のノートナンバーが入っている。

① 論理型配列 noteNumber[note]に false をセットする。

② 全ての鍵盤から指が離れたか isEmpty()メソッドで確認する。このメソッドは全ての配列の値が false になっている場合 true を返す。

③ 全ての鍵盤から指が離れていたらアニメーションを停止する。

④ 元の状態に戻す。

```
public void onMidiNoteOff(final MidiInputDevice sender, int cable, int channel, int note, int
velocity) {
        ～略～
        noteNumber.set(note,false);                         //① 押された鍵盤に対して false をセット
        if( noteNumber.isEmpty() ){                         //② 全ての鍵盤の状態をチェック
                ((AnimationDrawable)mImageAnimation
                .getBackground()).stop();                   //③ アニメーションの停止
                mImageAnimation.setBackground
                Resource( R.drawable.frame2_2 );            //④ 元の状態に戻す
}
```

図 4.6.2　onMidiNoteOff の処理内容

5 章　テスト

5.1　実施方法

函館高専の19～20歳までの男女6名（男2名、女4名）に、聴覚障害者と同じ条件にするため音を出さずにアプリケーションを操作してもらって音楽演奏を楽しめるかどうかテストを行った。

5.2　集計結果

アンケートの集計結果は表 5.1 の通りである。

(1) 音がなくても、アニメーションだけで演奏を楽しめたか？
楽しめた・・・4人　　楽しめなかった・・・2人
(2) アプリケーションとして使いやすいか？
使いやすい・・・6人　使いにくい・・・0 人
(3) 動作上の不具合はあったか？
ある・・・2人　ない・・・4人
(4) 意見や改善点があったら書いてください。
・動作が遅い（フィードバックの時間）。
・動作のバリエーションがもっとたくさんあるといい。
・ボタンそれぞれ違うパターンにしてほしい。
・涙が背景と同化してて残念。
・アニメーションは楽しめたが、演奏としては楽しめなかった。

表 5.1　アンケート集計結果 (テスト実施日：1月31（金）)

6 章　結果と考察

アンケートの集計結果より、アプリケーションを楽しめなかった原因として、アニメーションの種類が少ない、キャラクターの表現に限界があった、動作上に不具合があった、画面デザインが見づらかったという点が考えられる。

6.1　アニメーションの種類について

アニメーションの種類が少ないという意見については、アニメーションの種類が少なく物足りないとユーザーに感じさせてしまったことが原因であるので、その改善策として透明化や移動などを実現する Tween アニメーション[6]等を使用して、アニメーションの種類を増やす工夫が必要であると考えられる。ここで Tween アニメーションとは一枚の画像を用いて移動・拡大縮小・回転・透過などを組み合わせたアニメーションのことである。

また、キャラクターをもう一つ増やして両手での演奏に対応させる、背景の変化を取り入れるなどの手法も考えられる。例えば今回は12音階に対応した12パターンのアニメーションしかないので、前の音と現在の音の組み合わせを考慮してアニメーションのパターンを増やしていく等の方法が必要である。

6.2　キャラクター表現について

キャラクターの表現に限界があったという意見については、今回使用したキャラクターでは手足と表情以外の体の動きが表現しにくく、ステップなどの細かい動きが出来なかったことが原因であるので、その改善策としてより細かい動きで踊っているよう見せるために人間をモチーフにしたキャラクターをデザインする必要がある。

6.3　動作上の不具合について

動作上に不具合があった点については、キーボードを押してからアニメーションが始めるまでの時間が遅かったことが因であるので、プログラムを見直す必要がある。

6.4　デザインについて

画面デザインが見づらかった点については、背景とアニメーションの一部が同化して見づらかったことが原因であったと思うので、その改善策として背景とキャラクターの区別をはっきりさせる必要がある。

7章　総括

7.1　まとめ

MIDI Driver [4]を使用して聴覚障害者が音楽演奏を楽しむことが出来る Android アプリケーションを開発した。キーボードを演奏したときに画面に表示されるアニメーションと、タイトル画面、使い方、背景画面を作成し、MIDI Driver のメソッドを利用してアニメーションを実装した。さらにアプリケーションのテストを行い改善点をまとめた。

7.2　今後の課題

MIDI Driver の解析が及ばず終了処理に関する不具合を改善出来なかったので、もう少し時間をかけて内容を解析し、アプリケーションのパフォーマンスを向上させる。また、テストの際に得られた意見を参考して、音楽演奏を楽しめるようにアニメーションの表現を広げていく必要がある。

なお、本アプリケーションは 2014 年 3 月 8 日に函館高専で開催される「ものづくり成果の体験・展示会」において展示される予定である。

参考文献

[1] 障害者の日常が IT で“革命的”に変化
http://www.nikkeibp.co.jp/article/news/20131009/368354/?rt=nocnt

[2] Android アプリ開発における Frame アニメーション
http://monoist.atmarkit.co.jp/mn/articles/1205/21/news003.html

[3] 表情35
http://www.pixiv.net/member_illust.php?mode=medium&illust_id=3048401

[4] Android USB MIDI Driver のご紹介 | kshoji
http://blog.kshoji.jp/2012/11/android-usb-midi-driver.html

[5] 音楽研究所 MIDI ノート番号と音名、周波数の対応表
http://www.asahi-net.or.jp/˜hb9tktd/music/Japan/Research/DTM/freq_map.html

[6] Android アプリにおける Tween アニメーション
http://monoist.atmarkit.co.jp/mn/articles/1205/11/news003.hJ