

# 6点入力方式によるスマートフォンのメモ帳の実装

平山 亮<sup>†</sup> 田辺 敬典<sup>††</sup>

<sup>† ††</sup>大阪工業大学情報科学部  
〒573-0196 大阪府枚方市北山 1-79-1

E-mail: <sup>†</sup>makoto.hirayama@oit.ac.jp, <sup>††</sup>e1c11056@st.oit.ac.jp

**あらまし** スマートフォンはキーボード・ボタンなどがディスプレイ上に表示されるだけであるため、ディスプレイを視認できない利用者は、文字入力が困難である。そこで長谷川貞夫氏が提案した点字を利用した6点入力による文字入力方式を使い、視覚障がい者でもスマートフォンを使ってスムーズに文字入力が行えることを目的とした簡易メモアプリケーションを試作した。

**キーワード** 視覚障がい者、文字入力、点字、スマートフォン、視覚障がい者支援

## A notepad application on smartphones using the six dots input method

Makoto J. HIRAYAMA<sup>†</sup> Keisuke TANABE<sup>††</sup>

<sup>† ††</sup>Osaka Institute of Technology University  
1-79-1 Kitayama, Hirakata-city, Osaka, 573-0196 Japan

E-mail: <sup>†</sup>makoto.hirayama@oit.ac.jp <sup>††</sup>e1c11056@st.oit.ac.jp

**Abstract** On smartphones, a keyboard and buttons are displayed as soft-keys on touch screens, thus, it is difficult for visually impaired users to input texts to notepad or Email applications. Thus, a prototype smartphone application of a memo pad, which supports the six dots input method proposed by Hasegawa, was implemented.

**Keyword** visually impaired persons, text input method, braille, smartphone, assistive technology

### 1. はじめに

視覚障がい者や盲ろう者が、スマートフォンを用いて、情報取得やコミュニケーションを行う際、文字入力が必要であるが、タッチパネルのディスプレイでは、キーボードやボタンはソフトキーであるため、資格障害者は視認できず、入力に苦勞する。解決のための事例としては、視覚障害者のためのスマートフォンの開発<sup>[1]</sup>、文字入力方法を検討した研究<sup>[2]</sup>などが発表されている。今回の報告では、スマートフォンでの文字入力をよりスムーズに行うための点字を利用したスマートフォンにおける文字入力システムについて、長谷川貞夫氏による点字を応用した6点入力方法の提案<sup>[3]</sup><sup>[4]</sup><sup>[5]</sup><sup>[6]</sup>をまとめ、それに基づき、アンドロイドスマートフォンの簡易メモ帳を実装したことについて報告する。

### 2. 文字入力支援システムについて(文献[10]より再掲)

本稿にて紹介する点字を利用した文字入力支援システムの概要について説明する。

### 2.1. 点字について

これまで点字が視覚障がい者のコミュニケーションや教育に果たした役割は大きい<sup>[7]</sup>。

点字にはおおよそ200年の歴史があり、その原型は1822年にフランスの軍人 Charles Barbier が軍事機密保持を目的とし、声を出さずに命令を伝達する暗号として考案した縦6点・横2列の12点式点字方式(図1)であった<sup>[8]</sup>。

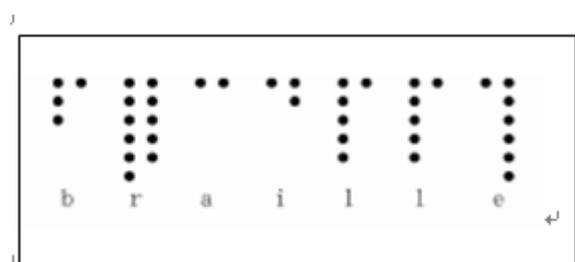


図1 Charles Barbier の12点式点字

現在の点字の構造は1829年に Louis Braille が Charles Barbier の12点式点字からヒントを得て縦3

点・横2列の6点式点字でアルファベットや数字の体系を完成させたことにより生まれた<sup>[8]</sup>。そして、この6点よりなる範囲をマスと呼び6個の点には図2のように1の点から6の点まで、順に名前がついている。

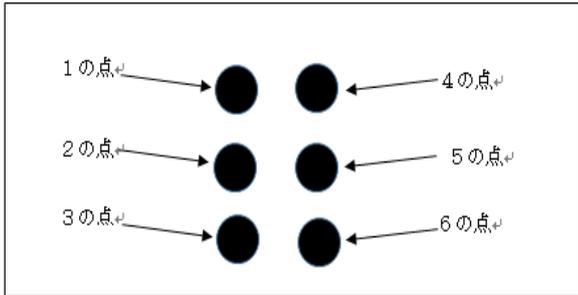


図2 各点の名称

この6個の点から作られる符号は63個であり、1個も点のないものを「マスアケ」と呼び一般のスペースのような働きをする。

## 2.2. 長谷川氏による提案の入力方法

テンキーを利用するより視覚障がい者や盲ろう者の多くがなじみのある点字を利用すれば誤字や脱字を減らし、よりスムーズに文字入力や情報発信を行うことができるのではないかと考えられる。あらかじめ五十音の点字の点の配置を2桁の8進数に対応させる。具体的には図2で示した1の点、2の点、3の点による点の配置状況で1桁を表現。残りの4の点、5の点、6の点による点の配置状況でもう1桁を表現(図3参照)。この二つの桁を組み合わせで五十音を表現する。

| 五十音    | あ   | お   | い   | れ   | ぬ   | み   |
|--------|-----|-----|-----|-----|-----|-----|
| 各列左より  | ●   | ●   | ●   | ●   | ●   | ●   |
| 1と4の点  | ● — | — ● | ● — | ● ● | ● ● | ● — |
| 2と5の点  | — — | ● — | ● — | ● ● | — — | ● ● |
| 3と6の点  | — — | — — | — — | — — | ● — | ● ● |
| 2桁の8進数 | 1 0 | 2 1 | 3 0 | 3 3 | 5 1 | 7 6 |

図3 点字と2桁8進数の対応例

なお、2桁の8進数が両方とも0の場合は前述した「マスアケ」として扱う。また、古文などで用いられる「ゐ」や「ゑ」は現代文で用いられる「い」や「え」と区別するものとする(図4)。

次に図3のように点字を基に対応させた2桁の8進数を向きと角度が決まった直線に対応させる考え方について記述する。直線は45度ずつの8本の線を考える。また、角度については入力画面上から上に垂直に上がる直線を0度とする。そして、直線の角度0度を8進数の0として考え、45度の場合1、90度の場合

2のように8種類の角度に0から7の8進数を割り当てる(図5)。

| 五十音    | い   | え   | ゐ   | ゑ   |
|--------|-----|-----|-----|-----|
| 1と4の点  | ● — | ● ● | — — | — — |
| 2と5の点  | ● — | ● — | ● — | ● ● |
| 3と6の点  | — — | — — | ● — | ● — |
| 2桁の8進数 | 3 0 | 3 1 | 6 0 | 6 2 |

図4 「い」、「え」、「ゐ」、「ゑ」の点字と2桁の8進数

| 直線の角度 | 角度に対応する8進数 | 補足            |
|-------|------------|---------------|
| 0°    | 0          | 上に垂直に上がる直線    |
| 45°   | 1          | 右上方向45度に上がる直線 |
| 90°   | 2          | 右方向の水平線       |
| 135°  | 3          | 右下方向45度に下がる直線 |
| 180°  | 4          | 垂直に下がる直線      |
| 225°  | 5          | 左下方向45度に下がる直線 |
| 270°  | 6          | 左方向の水平線       |
| 315°  | 7          | 左上方向に上がる直線    |

図5 直線の角度とそれに対応する8進数

実際に使用者が文字入力を行うときはタッチ画面上に入力した二本の直線の組み合わせで一文字が入力される。

## 3. 実装

### 3.1 開発環境

今回、Androidアプリケーションを開発するにあたって使用したソフトウェアや開発環境などを示す。

#### (1)JDK(Java開発キット)

Javaアプリを実行するためのJava仮想マシンをはじめ、コンパイラ、クラスライブラリ、デバッガなどを備えた開発キットである。

#### (2)Eclipse(総合開発環境)

オープンソースで提供されており、エディターやプロジェクト管理、デバッカーなど、おおよそのアプリ開発には欠かせない諸機能を取り揃えた開発環境である<sup>[9]</sup>(図6)。また、プラグインと呼ばれる拡張ライブラリを導入することで、自在にさまざまな機能を追加できる。

#### (3)Android SDK

Androidアプリを開発するのに必要なソフトウェア。基本的なクラスライブラリをはじめ、エミュレーター、ドキュメント、サンプルプログラム、ビルドやテストに必要なツールが一式含まれている。

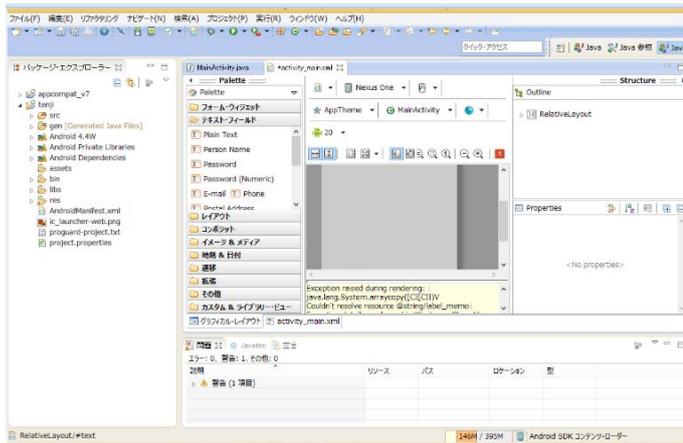


図 6 Eclipse 上における開発画面

#### 4. メモ帳アプリケーションのテスト

開発したアプリケーションは、アンドロイドのスマートフォンにダウンロードしてテストを行う。今回は Xperia SO-02E (docomo) でテストを行った。アンドロイドのバージョンは、4.2.2 である。

今回試作した簡易メモ帳のアプリケーションを立ち上げると、文字入力画面になる。6 点入力方式により、タッチ画面上で、2 本の直線を指でなぞって入力した後、ダブルタップすると、その文字入力を認識して画面上に文字が表示される。今回は実装を簡単にするため、2 本の直線を入力した後、ダブルタップした時点で文字を確定する仕様としたが、1 文字ごとにダブルタップが必要なので、将来の実装では、2 本目の直線が入力されたのを認識した時点で文字を表示するように改良する予定である。

入力方法は、例えば「あ」、「い」、「う」、「え」、「お」を入力したい場合、図 7 に示されている向き・角度で直線を入力すればよい。

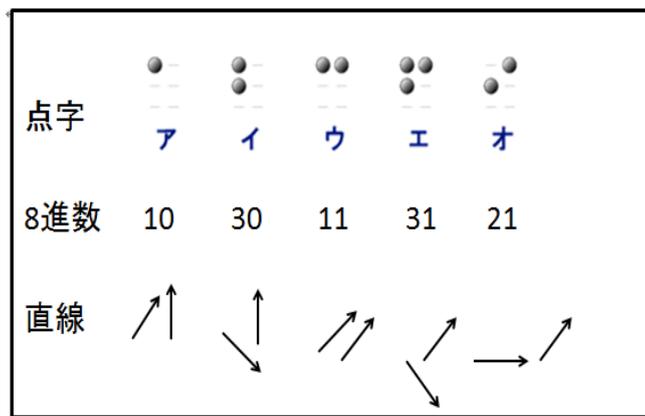


図 7 「あ」、「い」、「う」、「え」、「お」における点字とそれに対応した 8 進数と直線

実際に入力を試した時の、端末を図 8 に示す。画面構成の模式図は図 9 のようである。



図 8 アプリ起動画面

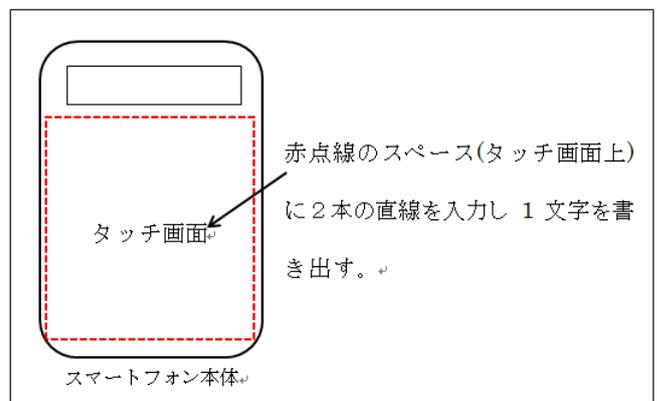


図 9 スマートフォンにおける 6 点入力方法

今回の実装では、文字入力が行われたときに、入力した文字を音声フィードバックする機能を追加して、入力された文字が視覚障害にもわかるようにしたが、不具合があり、音声フィードバック機能については、デバッグ中である。現在完成を目指して実装を進めている。

## 5. おわりに

6点入力による文字入力方式を用いて視覚障がい者でもスマートフォンを使ってスムーズに文字入力が行えることを目的とした簡易メモアプリを試作した。

現在は実装中で、簡単な文字入力ができるようになった状態だが、音声フィードバック機能を含めて、近々完成させ、実証実験を行って、評価をする予定である。

## 文 献

- [1] 釜江常好, 小出富夫, “視覚障害者用アンドロイド・スマホおよびタブレットの開発—職場への完全参加を目指して—”, 2014 年度年次大会企画セッション, 2014.
- [2] 橋本遼, 渡辺昌洋, 浅野洋子, “タッチパネル式インターフェースを持つ情報端末のアクセシビリティ”, 2012 信学技報, Vol. 112, No. 45, HCS2012-15, pp. 109-114, 2012.
- [3] 長谷川貞夫, 成松一郎, 武藤繁夫, 新井隆志, “ヘレンケラースマホの本質とロボットアームの展開”, 画像電子会 第4回視覚・聴覚支援システム研究会, 2014
- [4] 長谷川貞夫, “長谷川貞夫の視覚障害とユビキタスバリアフリー”, 「ワイヤレスジャパン 2013」と「ヘレンケラースマホの本質」, 2013.  
[http://ubq-brl.at.webry.info/201306/article\\_1.html](http://ubq-brl.at.webry.info/201306/article_1.html)
- [5] 長谷川貞夫, “これからは視覚障がい者がロボットアームで三次元映像の形を触知する技術が必要—コンピュータ利用の初期から点字で日本語処理を行ってきた視覚障がい者の立場から—”, 2014 年度年次大会企画セッション, 2014.
- [6] 長谷川貞夫, “スマートフォンとロボットアームの提案”, 私信, 2014.
- [7] 木塚泰弘 “中途視覚障害者の触読効率を 向上させるための総合的 点字学習システムの開発”, 1999.  
<http://web.econ.keio.ac.jp/staff/nakanoy/article/braille/BR/index.html>
- [8] 黒崎恵津子, 点字のれきし, 汐文社, pp. 25-39, 1998.
- [9] 山田祥寛, はじめての Android アプリ開発, 秀和システム, pp. 2-43, 2012.
- [10] 田辺敬典, 平山亮, 視覚障害者のスマートフォン利用を想定した6点入力方法の試作, 画像関連学会連合会第1回秋季合同大会, PC-01, 2014.