

Connected,Limited Device Configuration (CLDC) API
を使った i アプリ RPG ゲームの開発

平成 20 年 1 月 31 日
経営情報学部経営情報学科
5104258 藤田敬之

目次

第 1 章.はじめに	p 3
第 2 章.既存技術の説明	p 4
2.1. 開発実行環境のソフトウェア構成	p 5
2.2.i ppli Development Kit for JDK-4.0(FOMA)概要	p 5
2.3.エミュレータの利点	p 7
2.4.エミュレータによる i アプリの開発	p 7
第 3 章.関連研究	p11
3.1.構成	p12
3.2. IApplication 継承クラス (RPG2 クラス)	p12
3.3.キーイベント (RpgCnav)	p13
3.4. ScratchPad と ImageMap の使用 (MapData)	p14
第 4 章.考察	p17
第 5 章.まとめ	p18
第 6 章.参考文献	p19
謝辞	p20

第 1 章 はじめに

現在、携帯電話（携帯情報端末）はパソコンよりも広く普及しているのが現状である。また、2001 年 1 月下旬より携帯電話に、Java 実行環境を搭載した端末が発表されて以来、携帯電話は「パソコン化」の一途を辿ってきた。

そして現在、携帯電話を対象とした Web ページや Java アプリケーションは一層充実している。そこで現在、Java アプリケーション開発ということへの関心が高まり、開発ツールも初級者から上級者まで使いやすいものが求められると考えた。

今回、i アプリのエミュレータ制作を将来の目標とするため、現在提供されているエミュレータを使用することで、経験、知識、開発技術を得る事を現段階の目的とし、本プログラムを作成することにした。

エミュレータは NTT DoCoMo によって無償で提供されている、i ppli Development Kit for JDK-4.0(JDK)を使用する。JDK を選んだ理由として、他のキャリアのアプリを開発するよりも、書籍やサンプルプログラムなどを探するのが比較的容易であったからである。

2 章では既存技術を説明し、3 章ではエミュレータというものを知るために作成した「Over The Rainbow QUEST i アプリ版」について説明した。

第 2 章 既存技術の説明

i アプリ対応携帯電話 [1]には、Java 2 Platform Micro Edition (J2ME) の Connected, Limited Device Configuration (CLDC) がサポートされている。J2ME CLDC では、小型のネットワーク接続デバイス向けの Java 仮想マシン仕様およびクラスライブラリに関して必要最小限の機能を定義している。

CLDC [2]は、K Virtual Machine (KVM) をベースに作成されている。KVM は、携帯電話のようにメモリや CPU、消費電力に制約があるデバイス向けに設計されたコンパクトで移植性の高い Java Virtual Machine である。KVM は、Java2 ファミリにおける上位エディションの Java 実行環境に含まれる機能を多く共有しているが、小型デバイスの要求を満たすために、小型デバイスの制約に合わせて最適化されている。よって携帯電話という制約のあるデバイスでも、コンパクトであるとはいえ、必要な Java Virtual Machine やクラスライブラリを備えているので、充実した i アプリを開発提供することが出来る。

CLDC については、サン・マイクロシステムズの下記の Web ページより、ライブラリ仕様を入手できる。

<http://www.nttdocomo.co.jp/i/java.html>

2.1. 開発実行環境のソフトウェア構成

i アプリ実行環境は、次の API レイヤから構成されている。(図 1 参照)

1. J2ME/CLDC で規定された API
多様な小規模デバイス用のアプリケーション開発およびプロファイル定義のために、最小限の有用なライブラリのセットを目的とする。
2. i アプリ API (i アプリ基本 API)
CLDC を土台とした、i アプリサービスに特化したアプリケーション基盤を提供し、全機種に共通的に搭載される API。
3. i アプリ API (i アプリオプション API)
搭載有無の判断がメーカーに委ねられる API。3D グラフィックス描画やサブディスプレイ描画などがある。
4. i アプリ API (i アプリ拡張 API)
API および動作がメーカー毎に異なる可能性のある API。指紋センサーや FeliCa チップなどがある。

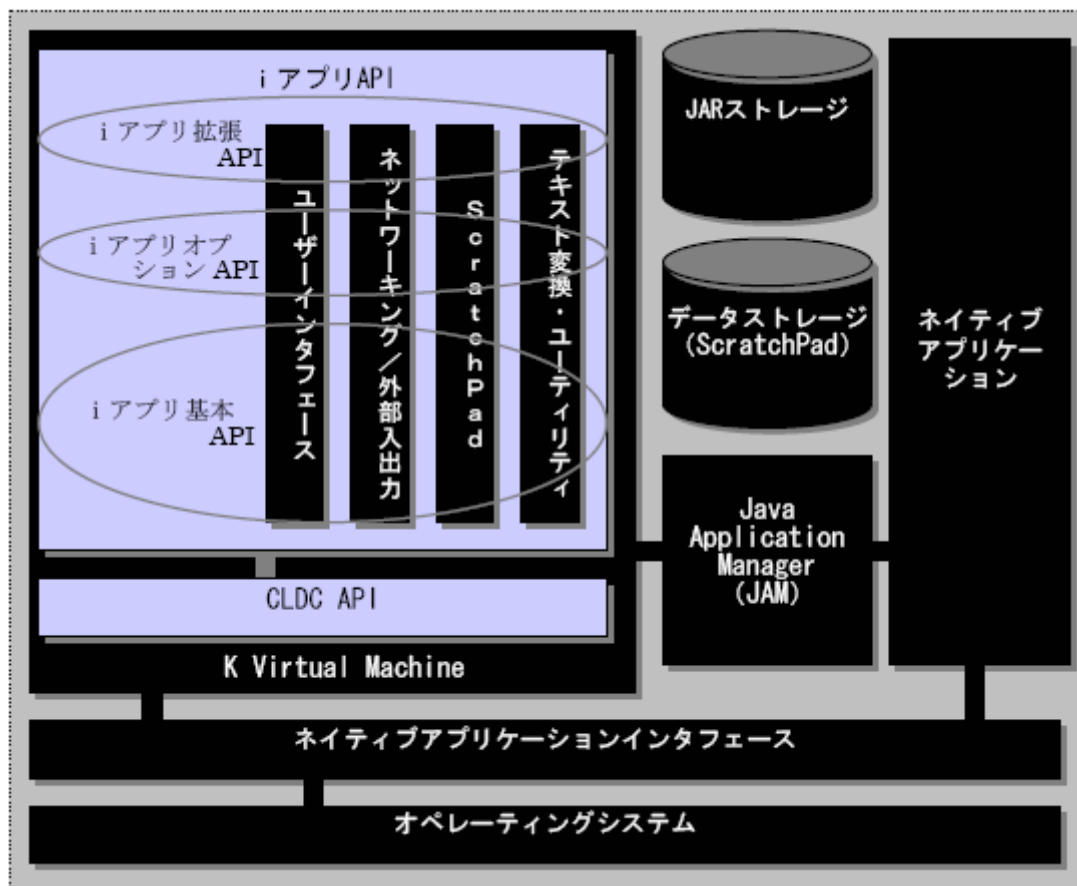


図 1 . i アプリ対応携帯電話における i アプリ実行環境

2.2.i ppli Development Kit for JDK-4.0(FOMA)概要

どの機種をターゲットにするかによって使用するエミュレータ(i ppliTool)も変わってくる。今回使用した“i ppli Development Kit for JDK-4.0”[3]は JDK-4.0 プロファイル向け i アプリの開発をサポートするためのツールであり、i アプリのビルドから PC 上でのエミュレートまでをサポートしている。

“i ppli Development Kit for JDK-4.0”から新規に追加された i アプリオプション API の機能は以下の通りである。

- アロケート可能な連続メモリ量の参照
i アプリ実行環境の確保可能なメモリ領域上で、現在どの程度のサイズ連続したメモリバッファを確保できるかを参照する。
- オーディオ出力のサラウンド制御
サラウンド機能をサポートしている携帯において、i アプリからのサウンド再生時のサラウンドのオン、オフを制御している。
- 携帯電話ネイティブのデータストレージアクセス

待ち受け画像など、携帯電話のネイティブアプリケーションが使用するデータストレージ（ ）にiアプリがアクセスする機能。

iアプリからアクセス可能な領域は、データストレージの中でiアプリ用に用意された一部分だけである。

i ppliTool は NTT ドコモのオフィシャルウェブサイトより以下の手順でダウンロードできる。

1. NTTドコモ ホーム
2. iモード
3. 作ろうiモードコンテンツ
4. iアプリ
5. iアプリ開発ツールのダウンロード

<http://www.nttdocomo.ne.jp>

iアプリの対応端末としては、901iシリーズ向けのツールである。JDK-3.x プロファイル以前のプロファイル向けに作成したアプリケーションは、一部の機種依存の振舞いを除いて、従来通り JDK-4.x プロファイル搭載携帯電話で動作することができる。(図2 参照)

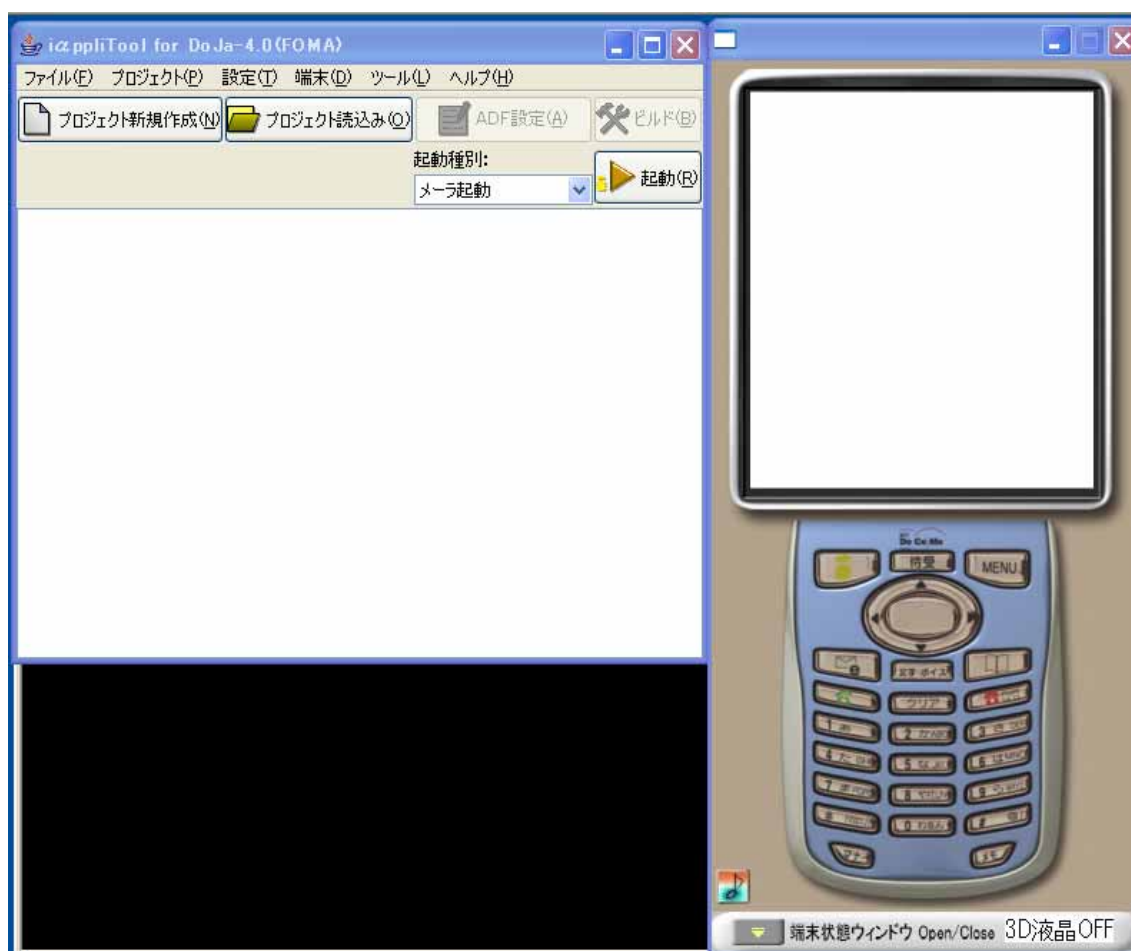


図 2 . i ppli Development Kit for JDK-4.0

JDK-4.x プロファイルでは、以下のような機能が強化されている。

- 3D 描画機能の強化
JDK-4.x プロファイルでは com.nttdocomo.ui.graphics3d パッケージ[4]に、アプリ基本 API として 3D グラフィックス描画機能が追加されている。
- 3D サウンド機能の実装
携帯電話に搭載されているステレオスピーカを利用し、さまざまな方向から立体的にサウンドが聴こえるような効果を与えることが出来る機能。
- CLDC1.1 をサポート
浮動小数点の利用が可能等。

2.3. 統合開発環境

統合開発環境 (IDE) は、アプリケーションの作成に必要な編集、コンパイル、実行、デバックの機能が統合されている。“i ppli Development Kit for JDK-4.0” では、Sun ONE

Studio 4, Mobile Edition および Eclipse で i アプリ開発するために必要な拡張モジュールを提供している。これらの統合開発環境で、i アプリの編集、コンパイル、ビルド、実行、デバックまでを一貫して行うことができる。

- Sun ONE Studio 4 : <http://docs.sun.com/app/docs/coll/1019.2?l=ja>
- Eclipse : <http://www.eclipse.org/>

2.4. エミュレータによる i アプリの開発

JDK では、アプリケーションを“プロジェクト”という単位で管理し、コンパイルやアプリケーションの配布に必要な ADF ファイルの作成などを行ってくれる。

i アプリの開発サイクルを次に記す：

1. i アプリのプロジェクト作成
2. i アプリのプログラム (.java) を作成
3. i アプリで扱うリソース (画像や音楽等) を設定
4. アプリケーションディスクブリタファイル (ADF) を設定
5. プロジェクトのビルド

エミュレータでビルドを実行すると次の作業がまとめておこなわれる。

- コンパイル
J2SE SDK コンパイラを使用して、ソースファイルをクラスファイルにコンパイルする。
 - バイトコードの検証
事前検証ツールを使用して、Java クラスの事前検証をする共に、Java クラスの中で浮動少数点演算やファイナライズメソッドが使用されていないかどうかを検査する。
 - パッケージング
J2SE SDK の jar ツールを使用して、事前検証された全てのクラスファイルとソースファイルを JAR 形式にパッケージングする。
 - ADF の作成
i アプリの情報を記述するテキストファイルを作成する。
 - ダウンロード HTML の作成
作成した i アプリをダウンロードするためのもっともシンプルな HTML を作成する。
6. エミュレータで実行
エミュレータ上で i アプリを実行する。

もし、i アプリの動作に問題があれば 2. からやり直す。

7. i モード携帯電話で実行

作成した i アプリ一式を HTTP サーバ上で配置後、i モード携帯電話で i アプリをダウンロードして実行する。

上記サイクルをフロー図に示す。(図 3 参照)

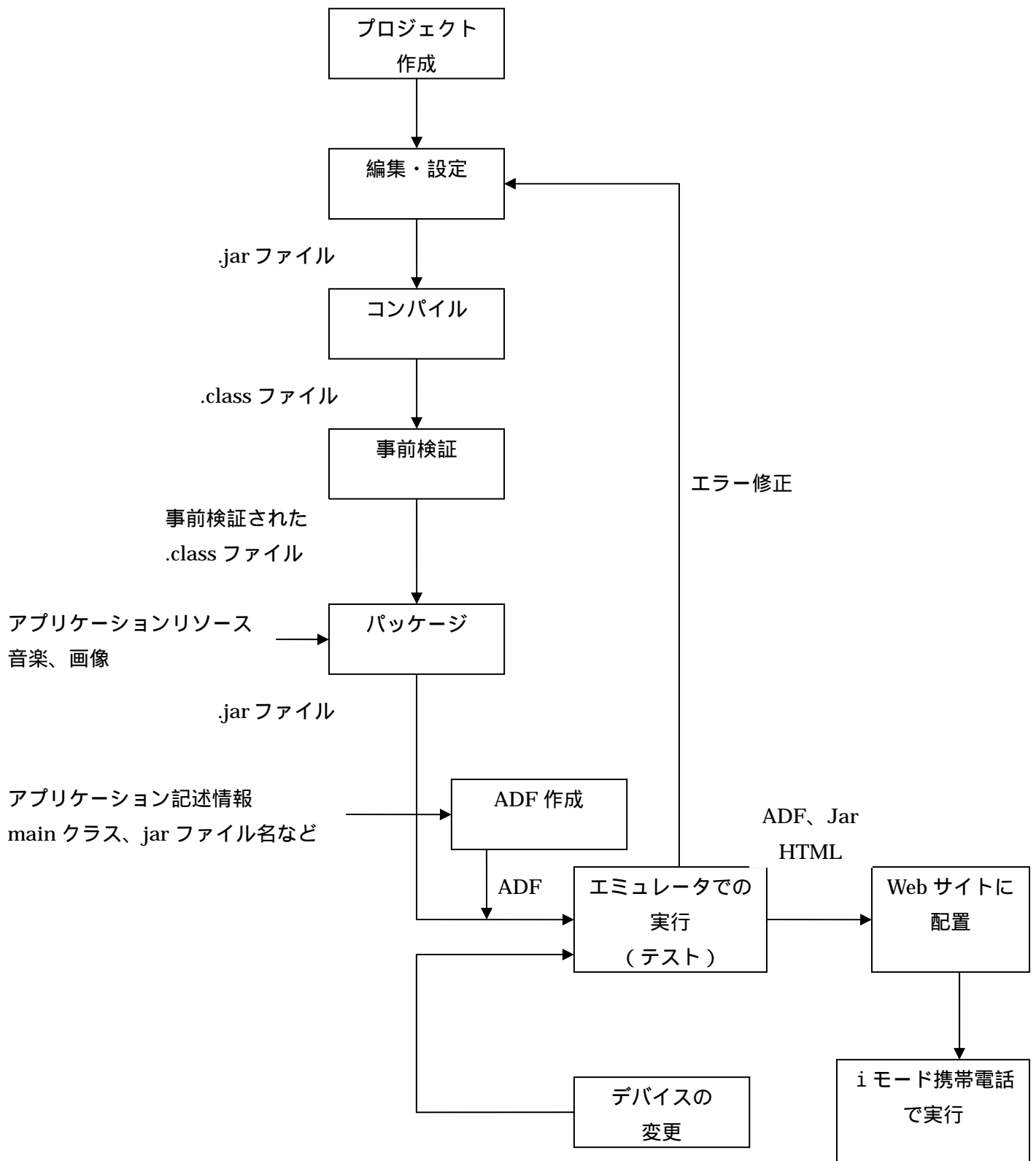


図 3.アプリケーション開発サイクル

第 3 章 関連研究

本プログラムは、「Over The Rainbow QUEST」という先輩の製作したゲームを i アプリに移行することを目的とした。(図 4 参照)



図 4.Over The Rainbow QUEST

背景やキャラクターは、先輩が photo shop でデザインした image があったので、それを流用させてもらった。プログラムは、先輩の「Over The Rainbow QUEST」がパソコン上で動作することを目的に作られたソースであったため、JDK の「i モード Java 拡張 API」下故に起こるエラーが膨大であった。

そこで今回は、始めノートパッドで作成し、その後、統合開発環境 (Eclipse) を用いて一からプログラムを構築した。

3.1.構成

本プログラムは、次のクラスによって管理されている。

- RPG2 クラス
アプリケーション起動時に最初に呼び出される IApplication クラスを継承している。
- RpgCanv
描画とプレイヤーの動作を管理している。
- MapData クラス
Map データを管理し Canvas クラスを継承している。
- Player クラス
プレイヤーデータの管理を行っている。

その他に画像や音楽、バイナリデータ等を扱うフォルダがある。

- bin
JAR/ADF/HTML のフォルダ
- res
リソースファイルのフォルダ
- src
ソースファイルのフォルダ
- sp
スクラッチパッドのフォルダ
- class
クラスファイルのフォルダ

3.2.IApplication 継承クラス (RPG2 クラス)

図 6 は、i アプリを作る上で、必要な IApplication クラスを継承したクラスのソースである。

```
public void start() {  
    // TODO 自動生成されたメソッド・スタブ  
    canvas = new RpgCanv();  
    Display.setCurrent(canvas);  
}
```

図 6 . IApplication 継承クラスのソース

る。IApplication 継承クラスには start というメソッドがあり、i アプリ起動時に、一番初

めに呼び出されるクラスである。

Display.setCurrent(canvas)で、起動時のクラスを呼び出している。canvas = new RpgCanv();なので、RpgCanv クラスが呼び出されている。

3.3.キーイベント (RpgCnav)

Canvas 継承クラスでは、低レベル API でのイベント処理を扱う。イベントは processEvent メソッドをオーバーライドし、イベント発生時に、イベント種別とそれに伴うパラメータが渡される。(表1参照)

イベント種別	パラメータ
KEY_PRESSED_EVENT	押されたキーの値
KEY_RELEASED_EVENT	離されたキーの値
TIMER_EXPIRED_EVENT	タイマーイベントの発生した ShortTimer のタイマーID

表1 .低レベル API

```
// イベント応答
//          ImageMap の移動
public void processEvent(int type,int param){
    if(type==Display.KEY_PRESSED_EVENT){
        switch(param){

            case Display.KEY_UP:
                /* 下方向キーを押した時の処理 */
                ply.plyY = ply.plyY -15;
                map.orgY = map.orgY +15;
                repaint();
                break;

                省略

            case Display.KEY_RIGHT:
                /* 右方向キーを押した時の処理 */
                ply.plyX = ply.plyX +15;
                map.orgX = map.orgX -15;
                repaint();
                break;
        }
    }
}
```

図7では、KEY_PRESSED_EVENT でキーが押されたとき、キャラクターが移動するようになっている。省略部分の上方向キーと左方向キーを押したときの処理は、それぞれ+と-の値を逆にするだけである。case 文にしたのは、キー押しっぱなし処理を行うときに

```
int mapData[] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, //0
                 0,1,3,3,3,3,3,3,3,3,3,3,2,0, //1
                 0,7,9,9,9,9,9,9,9,9,9,9,8,0, //2
                 0,7,9,9,10,9,9,9,9,9,9,9,8,0, //3
                 0,7,9,9,9,9,9,9,9,9,9,9,8,0, //4
                 0,7,9,9,9,9,9,9,9,9,9,9,8,0, //5
                 0,7,10,9,9,9,9,9,11,9,10,9,9,8,0, //6
                 0,7,9,9,9,9,9,9,10,9,9,9,9,8,0, //7
                 0,7,10,9,9,9,9,9,9,11,9,9,9,8,0, //8
                 0,7,11,9,9,9,9,9,9,10,9,9,9,8,0, //10
                 0,7,9,9,9,9,9,9,9,9,9,9,9,8,0, //11
                 0,7,9,9,9,9,10,9,9,9,9,9,9,8,0, //12
                 0,7,9,9,9,9,9,9,9,9,11,9,9,8,0, //13
                 0,4,5,5,5,5,5,5,5,5,5,5,6,0, //14
                 0,0,0,0,0,0,0,0,0,0,0,0,0,0, //15
```

図8 . ImageMap のデータ

スムーズだからだ。

3.4. ScratchPad と ImageMap の使用 (MapData)

ScratchPad とは、i アプリから利用できるバイト配列で構成されたデータ保存領域のことである。携帯電話上の不揮発メモリを使用しており、電源を On・Off にしてもデータは残る。

本来 ImageMap というメソッドを使用したとき、図8のように記述しなければならない。しかし、今回のようにバイトデータを ScratchPad に格納することによって図9だけの記述ですむようになる。また、フリーのバイナリエディタを使用して簡単にデータの書き換えや作成ができる。

図 10 は、図 8 のバイトデータを格納し JDK のビューアーで表示したものである。

```
int mapData[]= new int[16*16];
```

図 9 . ImageMap 表示のソース

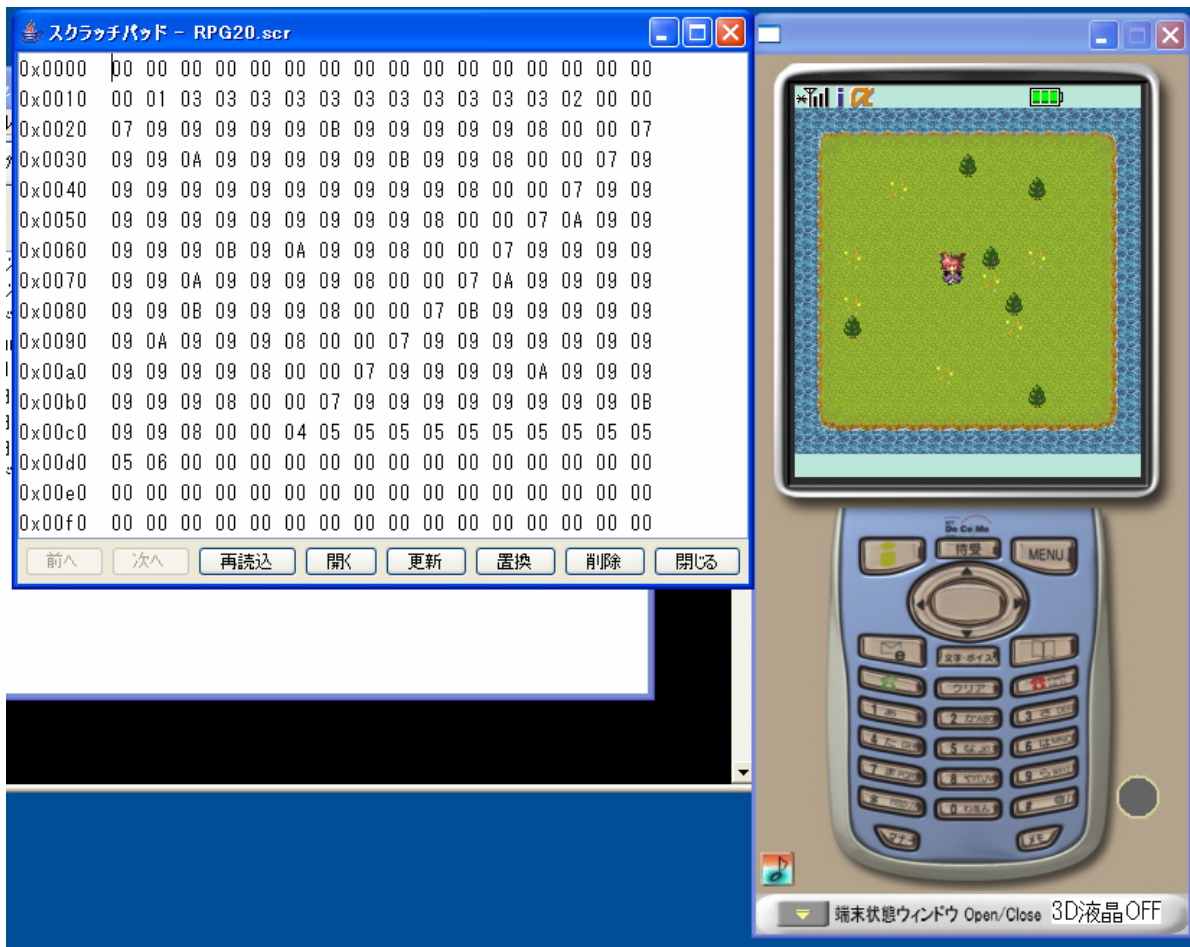


図 10 . バイトデータの格納と表示

第4章 考察

今現在における携帯の普及は目を見張るものがあり、尚且つ携帯電話の性能の向上により、様々なアプリが開発されているのが現状である。そのアプリだが、従来のアプリは開発者と呼ばれる専門者がいたのだが、一般ユーザにも開発が可能な時代となった。これに伴い様々なユーザがオリジナルゲームの開発を行い、独特の発想を持ったアプリの開発が行える時代にするべきだと感じた。

そこで実際、i ppli Development Kit for JDK-4.0 (FOMA) を使用して「Over The Rainbow QUEST i アプリ版」を開発し、比較的容易に開発することができた。しかし、まだ一般ユーザには容易に開発できないと感じる問題点が何点か存在した。それらの理由を述べる。

まず、JDK だが、これには開発環境が搭載されていないがために、統合開発環境のない開発者は、ノートパッドなどで開発していくことになる。総合開発環境の導入を行えばよいのだが、一般ユーザにそごう開発環境の導入は少し煩雑さを感じた。またノートパッド等のテキストエディタ等での開発も可能だが、JDK でビルドし、そこで出たエラー処理ということを繰り返してアプリケーションを作成しなければならない。これでは、簡単な i アプリすらも開発にとても時間がかかってしまう。

次に、ScratchPad は ADF 設定で領域確保をしてからでない、「プログラムにエラーはなく、ScratchPad も作成した。しかし実行することができない。」ということになる。また、RPG ゲームで背景の表示位置を ScratchPad に格納する場合、バイナリエディタを使っていたとしても、バイナリであるがために、実際に書いている値が合っているかわからない状態での作業となるので、いざ実行すると、実行結果にズレが生じるなど、統合開発環境で補えない部分もあった。

これらの問題点を改善した開発ツールを作成すれば、更なる i アプリ開発者の拡大から利用者の増大につながると考えられる。

第5章 まとめ

現在、携帯電話は「パソコン化」が進んできており、非常に高性能な電話へと変化してきている。そこで今回、その注目されるべき携帯電話に着目し、iアプリの開発を実際に行いたいと思い開発を行ってきた。更には実際に自分がパソコン上で作ったゲームが携帯電話でも動作する事が可能となるとすると、どのようなユーザでも容易に開発することができ、携帯電話業界は更に活性化すると推測した。

しかし、実際問題では、一般ユーザにiアプリの開発はまだ困難であると感じた。これからのiアプリの普及を考えるならこのような煩雑さを取り除いたiアプリの総合環境が必要だと感じた。

先に携帯電話はパソコン化してきていると述べたが、今後の携帯電話は容易に開発する事が可能となれば、パソコンで配布されているフリーソフトのように数多く、またゲームだけではなく、便利なアプリケーションの配信等が普及してくるであろう……。

前半部分では、携帯電話でのJavaの制約に触れ、JDKの概要から開発手順までを説明した。関連研究では、今回作成したアプリケーションの構成と使用した技術の説明をした。

関連研究では、実際にJDKを利用したことで開発環境の問題点を見出し、これからのiアプリのエミュレータ開発に活かせる改善点を考えることができた。

第 6 章 参考文献

[1] 株式会社 NTT ドコモ社：i アプリコンテンツ開発ガイド for DoJa-4.x/4.x LE
~ 詳細編 ~ ， 10P ， はじめに ， 2006 ．

[2] 株式会社 NTT ドコモ社： i アプリコンテンツ開発ガイド for DoJa-4.x/4.x
LE ~i アプリオプション・i アプリ拡張編~ ， 9P ， 2007 ．

[3] 株式会社 NTT ドコモ社：DoJa - 4.0 i ppli Development Kit ユーザーズガイド ，
12P ， 2005

謝辞

最後に、多くのことを学ばせてくれた花川先生に感謝します。本論文作成にあたり多くのご迷惑をおかけした、尊敬する尾花先輩に感謝します。家族に感謝します。卒業論文作成を応援してくれた川村優季さんに感謝します。同ゼミのみんなに感謝します。遅くまで研究室に残り、共に頑張っていた松谷栄輝君に感謝します。多くのアドバイスと勇気をあたえてくれた野口真司氏に感謝します。論文作成が行き詰った時に、励ましてくれた今井亮佑氏、守山洋輔氏に感謝します。