

2002 年度 卒 業 論 文

**3次元画像表示機能を持つ携帯電話におけるコ
ンテンツ作成を支援するツールキットの構築に
関する研究**

指導教員：渡辺 大地

メディア学部 Web3D プログラミングプロジェクト

学籍番号 99p083

大久保 隆

2003年3月

2002 年度 卒 業 論 文 概 要

論文題目

3次元画像表示機能を持つ携帯電話におけるコンテンツ作成を支援するツールキットの構築に関する研究

メディア学部 学籍番号: 99p083	氏 名	大久保 隆	主査	渡辺 大地
			副査	和田 篤

キーワード 携帯電話、Java、3次元、ツールキット

近年、Java 仮想マシンを搭載した Java 対応携帯電話が普及し、携帯電話上で動く 3次元コンテンツを開発することができるようになってきた。3次元コンテンツを作成するにはグラフィックスエンジンを使用するために3次元グラフィックスAPIを用いる必要があるが、そのようなAPIは概して習得が困難であり、初心者が3次元コンテンツを作成するのは容易ではない。また、APIの習熟者にとっても高度な数学的知識を必要とする複雑な形状の操作は難解であることが多く、開発効率を著しく下げる要因となっている。それらの問題を解決するための3次元グラフィックスツールキットはパーソナルコンピュータ用には存在するが、携帯電話用のツールキットは携帯電話において3次元コンテンツが作成できるようになってまだ間もないため存在しない。そこで、携帯電話用の3次元グラフィックスツールキットの構築を提案する。わかりやすいプログラミングインタフェースを提供し、複雑で難解な形状の制御に必要な高度な数学的知識と作業を開発者が意識することなく、自由に形状を制御できる機能を提供する。

目次

1	はじめに	1
2	ツールキットの概要	3
3	Java 対応端末における 3 次元画像表示機能の現状	
3.1	Java 対応携帯電話の現状	4
3.2	i アプリにおける 3 次元コンテンツ	6
4	ツールキットにおける工夫	
4.1	ツールキット本体のサイズ削減	7
4.2	移植性の保持	7
5	ツールキットの持つ機能	
5.1	ベクトルと行列	8
5.2	光源とマテリアル	8
5.3	形状表現	9
5.4	モデル	9
5.4.1	座標系	10
5.4.2	カメラの制御	11
5.4.3	拡大率とマテリアル	12
5.5	シーン	12
5.6	キャンバス	12
5.7	実際の記述例	13
6	考察・検証	
6.1	ツールキットに関する考察	14
6.2	ツールキットの課題	15
6.3	ツールキットの将来性	16

7	まとめ	17
---	-----------	----

謝辞

参考文献・URL

付録（FKME ツールキット仕様書）

1 はじめに

近年 Java 仮想マシンを搭載した携帯電話が普及し、携帯電話上で動くコンテンツを開発することができるようになってきた[1]。また、3次元画像を表示することができるグラフィックスエンジンも開発され、3次元で表示されるコンテンツを利用することができるようになってきた。携帯電話用 Java アプリケーション[2]の一つである i アプリ[3]において3次元コンテンツを開発するには、3次元グラフィックスエンジンである「Z3Dグラフィックスエンジン[4]」か「マスコットカプセル[5]」を用いる必要がある。後述するように Z3D グラフィックスエンジンのほうが3次元画像表示機能において優れているため、本研究では Z3D グラフィックスエンジンを使用している。Z3D グラフィックスエンジンを使用するには専用の API 群である「Z3D グラフィックスライブラリ[6]」を用いる必要がある。Z3D グラフィックスライブラリのコマンド体系はパーソナルコンピュータ上で3次元グラフィックスを実現する OpenGL API[7]と類似しているため、OpenGL の利用経験がある開発者には開発がしやすい。しかし、OpenGL はプリミティブな形状の制御しかできないため、開発者にとって形状を自由に扱いにくいのが難点である。Z3D グラフィックスライブラリも同様の問題点を抱えている。例えば、ある一点の方向を向いたままの回転や、形状を継承したときの回転の制御などは、毎回どの程度の回転量が与えられ、どの程度の平行移動を行うかを、複雑な座標変換などを用いて開発者が数値を導き出して与える必要があり、非常に難解かつ複雑で、頭を悩まされる作業である。プリミティブに形状の自由な移動を実現するには高度な数学的知識を必要とし、多くの時間を作業に費やすことになる。これは開発効率を著しく下げる要因である。

Z3D グラフィックスライブラリだけではなく、多くの3次元描画プラットフォームでは簡単な形状の表示をするだけのプログラムでもかなり多くのプログラムソースの行数を必要とする[7,8]。ただ直方体が回転しているだけのプログラムを書くだけでも数百行に達することも珍しくない。特に i アプリでは後述するように、30KB というアプリケーションサイズの制限があるため、プログラムソースのサイズが大きくなることはサイズ制限上好ましくない。また、i アプリはイベントハンドラ型であるため、リアルタイムにインタラクティブな操作が可能な3次元アニメーションを実現するには、毎回タイマーイベントを発生させてタイマーイベント中に値の変更を記述し、描画を行うという手順を踏まなければならない。これは単純にプログラムソースの行数を増加させると

いうだけではなく、直感的なリアルタイムグラフィックスアプリケーションの開発を行う妨げとなる。また、タイマーイベントとキーイベントが同時に呼び出された場合、キーイベントが無視されてしまうという問題も起きる。これはコンテンツの操作においてユーザビリティの低下をもたらす要因となる。

これらの問題は作り手にとって、コンテンツの内容といった本質的なものではない部分に余計な労力を要することを意味する。一般的にパーソナルコンピュータでは、前述した問題点を解決するために様々な 3 次元グラフィックスツールキットが開発されている[14-20]。ここでのツールキットとは、プログラミングインタフェースとして様々な機能を提供し、開発の効率化を実現するクラスライブラリ群のことを指す。描画ルーチンや様々な演算をクラスライブラリとしてカプセル化し、簡単なメソッドの組合せだけで 3 次元コンテンツを開発できるようにするものである。

3 次元コンテンツを開発する際に重要となる、複雑な操作や計算を簡易化してくれるツールキットはパーソナルコンピュータ用には存在するが、携帯電話の 3 次元コンテンツを開発できるようになって間もないため、携帯電話上で動作する 3 次元グラフィックスツールキットは、まだ存在しなかった。本研究では携帯電話用ツールキットを実際に構築し、その有用性を示す。

ツールキットの構築にあたって、前述した問題点を解決するために以下の点を念頭に置いて作成した。

強力かつ簡易な形状の制御: Z3D グラフィックスライブラリでは形状に対して行える制御はプリミティブな平行移動と軸回転移動のみである。これでは直感的な形状の制御が難しい。そこで複雑な形状の自由な制御を簡易に実現できる機能を提供する。

簡易な開発インタフェース: わかりやすい開発インタフェース (API) の提供により、初心者でも簡単に 3 次元コンテンツ作成ができるようにする。後述する 4 つのレイヤーの組み合わせと、イベントハンドラ型をループ処理型にすることにより開発効率の向上を図る。

プログラム行数の削減: 上述した開発インタフェースの使用により、後述する i アプリのサイズ制限と開発効率において重要な要素となるプログラムソースの行数の削減

を実現する。

本論文の章構成は次のとおりである。2章ではツールキットの概要を説明する。3章ではJava対応端末における3次元画像表示機能の現状を示す。4章ではツールキットの工夫点を示す。5章ではツールキットの機能を示す。6章と7章に考察とまとめを示す。

2 ツールキットの概要

ツールキットはFKシステム[15]をベースとし、携帯電話用Javaプラットフォーム上構築にした。このツールキットをFine Kernel Mobile Editionツールキット（以下FKMEツールキット）と名づけた。FKMEツールキットは、Z3Dグラフィックスライブラリを基盤とし、内部の処理や様々な演算を携帯電話用に修正したものである。

FKMEツールキットはFKシステムの理念を引き継いでおり、基盤となるレイヤーとして、「形状」、「モデル」、「シーン」、「キャンバス」の4つのレイヤーを持っている。形状はモデルに、モデルはシーンに、シーンはキャンバスにそれぞれ登録される仕組みになっている。以下にそれぞれのレイヤーについて説明する。

「形状」: 各種形状データを表す。FKMEツールキットでは、現状として直方体、球、線分、平面をサポートし、変数を定義するだけで作成することができる。また、VRMLモデルデータとして、一般的なwrl形式のVRMLファイルを三菱電機が開発した専用のコンバータツールで変換した際に出力されるmfg形式のモデルデータを保持することができる。また、同様に出力されるmtx形式のテクスチャデータをVRMLテクスチャデータとして保持することができる。mfg, mtx形式はZ3Dグラフィックスライブラリがサポートしている形式である。光源も形状データとして扱われ、平行光源、点光源などの光源の種類を定義できる。

「モデル」: 「モデル」は形状データとともに位置や方向情報、マテリアルといったステータスを保持するものである。形状の制御は、「形状」をモデルに登録することで実現する。「形状」と「モデル」はそれぞれ独立しており、「形状」の入れ替えによる変形アニメーションや、同じ形状を複数のモデルに登録することが可能である。

「シーン」: 一般的にディスプレイリストと呼ばれるもので、「モデル」を登録する機能をもつ。「シーン」は画面に描画するための複数の「モデル」と視点を構成する一つのカメラから構成され、カメラは一つの「モデル」で表現される。カメラは設定されている「モデル」を操作することで制御可能である。カメラには通常、専用の「モデル」を設定するが、他の「モデル」をカメラに設定することも可能で、一つの「モデル」が向いている場所に視点を切り替えることも瞬時に行うことができる。

「キャンバス」: 携帯上で3次元グラフィックスを実現するためには不可欠な、Z3DグラフィックスライブラリのクラスであるC3DCanvasを継承したものである。「シーン」を登録することで携帯電話の画面上に、「シーン」に登録されている「モデル」を描画する。一つの「キャンバス」には一つの「シーン」を描画することができる。「キャンバス」と「シーン」も独立しており、「キャンバス」に描画される「シーン」も入れ替えることが可能である。

FKME ツールキットは以上の4つのレイヤーをJAVAのクラスライブラリという形で実現している。本研究の目的の一つである簡易な開発インタフェースの提供はこの4つのレイヤーの組み合わせにより実現される。

3 Java 対応端末における 3 次元画像表示機能の現状

3.1 Java 対応携帯電話の現状

近年、携帯電話の普及率が増大し、日本の全人口のおよそ半数の 6204 万人が携帯電話を持っている[9][10]。そのうち携帯電話上で Java アプリケーションを動かすことのできるサービスに対応している Java 対応携帯電話は、全体のおよそ 4 割の数を占めている。Java アプリケーションは携帯電話会社により仕様が違い、NTT ドコモの i アプリ、J-PHONE の Java アプリ[11]、KDDI の EZPlus[12]がある。

また、Java 対応携帯電話の進化に伴い、Java アプリケーションにおいて 3 次元画像を表示することが可能になり、3 次元画像表示機能に特化した携帯電話も登場してきてい

る。Java アプリケーションにおいて 3 次元画像を表示するには、携帯電話に搭載されている 3 次元グラフィックスエンジンを使用する必要がある。現在、3 次元グラフィックスエンジンには、三菱電機が開発したハードウェアエンジンである、「Z3D グラフィックスエンジン」とエイチアイが開発したソフトウェアエンジンである「マスコットカプセル」の 2 種類がある。

「Z3D グラフィックスエンジン」は NTT ドコモの機種である三菱電機のムーバ D504i[13]のみに搭載されており、3 次元グラフィックス処理能力は毎秒最大 23 万頂点、520 万ピクセル、約 8 万ポリゴンである。8 つの光源による陰影計算、物体からの光の反射の計算、シェーディング、テクスチャマッピング、アンチエイリアシング処理、半透明処理といった処理をハードウェアで行うことができ、高画質な 3 次元画像を毎秒 8~15 フレーム程度で表示できる機能を持つ。

「マスコットカプセル」は NTT ドコモ、J-PHONE、KDDI の全機種に搭載されており、3 次元グラフィックス処理能力は、最大毎秒約 1 万ポリゴンと Z3D グラフィックスエンジンの 8 分の 1 程度である。Z 軸を定義できないため、物体のみを 3 次元で表示し、空間(背景)は 2 次元でしか表示することができない。光源は並行光源を一つ用意することができ、テクスチャマッピング、半透明処理を行うことができる。シェーディングはサポートしていない。描画フレーム数は毎秒 4~20 フレームである。

マスコットカプセルは現在バージョン 2 で、将来的には Z3D グラフィックスエンジンを超える 3 次元画像表示機能をもつバージョン 3 が登場するという。他にも Z3D グラフィックスエンジンを超えるハードウェア用グラフィックスエンジンの開発が行われているが、現在は Z3D グラフィックスエンジンと、マスコットカプセルバージョン 2 の 2 種類しかない。両者を比較すると、汎用性においては全機種に搭載されているマスコットカプセルのほうが優れているが、3 次元表示機能においては、描画ポリゴン数、シェーディングなど各種機能の面から見て Z3D グラフィックスエンジンのほうが遥かに優れている。

以上の点から、最も優れた携帯用 3 次元コンテンツを作成するには Z3D グラフィックスエンジンを使用するのが現在では最善の選択であり、本研究は Z3D グラフィックスエンジンを用いた 3 次元 i アプリコンテンツの開発を支援するツールキットの構築をするものとする。

3.2 i アプリにおける 3 次元コンテンツ

i アプリには通常のゲームなどに代表される通常アプリ、携帯電話の待ち受け画面において様々な処理を行う待ち受けアプリがあり、特に後者は近年注目を受けている。待ち受けアプリにより、携帯電話は秘書機能などエージェント的な役割を果たすことができる。例えば、普段は壁紙を表示し、一定の時刻になると自動的にネットワークに接続して、情報を取得するといった処理をアプリケーション側で制御することが可能となる。3次元画像表示機能に特化した携帯電話の登場により、i アプリはますます注目を受けている。

三菱電機では Z3D グラフィックスエンジンの機能を生かした D504i 向け 3 次元 i アプリコンテンツとして「どこでもチョコボ」を公開している。

この i アプリコンテンツは D504i にデフォルトでインストールされており、Z3D グラフィックスエンジンの性能を実感することができる。通常アプリである冒険編と、待ち受けアプリである待ち受け編が用意されており、特に冒険編は透過処理などを用いて見事な 3 次元空間を表現しており、携帯電話においての 3 次元描画の能力の高さを示すものである。待ち受け編は 3 次元壁紙として用いるが、冒険編との連携をとることができ、壁紙を自由に変えることが可能となっている。一例として「どこでもチョコボ - 待ち受け編」の画面イメージ図を図 1 に示す。



図 1：3 次元コンテンツ「どこでもチョコボ - 待ち受け編」

4 ツールキットにおける工夫

ここではツールキットを構築する際に行った工夫を記述する。

4.1 ツールキット本体のサイズ削減

今現在、i アプリのアプリケーションサイズは 30KB に制限されている。アプリケーションサイズは、プログラムソースと各種データを圧縮ファイルである JAR ファイルに圧縮したサイズで決定されるため、テクスチャやモデルデータを多用した大きなプログラムを作成することが難しい。アプリケーションごとにスクラッチパッドと呼ばれる保存領域があり、テクスチャデータやモデルデータを保存しておくことができるが、プログラムソースを保存しておくことはできない。

このサイズ制限はツールキットの開発の大きな障害となる。FKME ツールキットはクラスライブラリという形式上、i アプリ本体のサイズに含まれてしまうからである。そこで FKME ツールキットでは、ツールキット本体のクラスサイズを削減するために FK システムと比べて次のような工夫をした。

まず、クラス構成を簡略化し、ツールキット全体のサイズの削減を図った。FK システムの複雑なクラス構成のうち、4 つのレイヤーによる本質的な構成と、演算部のみを移植することによりクラス数を大幅に削減した。次に、メソッドを簡易化し、各クラスのサイズを削減した。携帯電話においては不要と思われるメソッドを削除し、必要な機能のみを作成することにより各クラスのサイズを削減した。

4.2 移植性の保持

FKME ツールキットの開発に際して、特に注意したのが FK システムとの移植性である。FK システムのメソッドと基本的に同じものを FKME ツールキットにおいて作成することにより、他開発環境上で、C++ 言語により FK システムを用いて開発したものを、大きく書き換えることなく携帯電話上で動かすことが可能となる。これにより携帯電話用エミュレーターの小さい開発画面ではわかりにくい詳細な動作なども、まず FK システムを用いてパーソナルコンピュータ上で開発してから、FKME ツールキットを用いて携帯電話用に移植するという手順を踏むことで、より洗練されたものを作ることができる。また、FK システムを使ったことのあるものならば、改めて Z3D グラフィックスライブラリを学ぶ必要がなく、3 次元コンテンツの開発を行える。

5 ツールキットの持つ機能

ここでは FKME ツールキットが持つクラスとその機能についての説明をする。

5.1 ベクトルと行列

FKME ツールキットはベクトルと行列演算機能を有する。クラスは「fk_Vector」と呼ばれるベクトルや座標を司るベクトルクラス、「fk_Matrix」と呼ばれる行列の演算、保持を司る行列クラスと、「fk_Angle」と呼ばれるヘッド角、ピッチ角、バンク角といったモデルの方向計算において有用なオイラー角を保持する角度クラスがある。

fk_Vector はベクトルの加減法、実数との積、内積、外積といった演算部を有し、座標の計算やモデルの制御の引数として利用される。

fk_Matrix は 4×4 の正方行列を保持し、回転行列の生成、平行移動行列の生成、拡大縮小行列の生成、オイラー回転行列の生成などの演算部を有する。

fk_Angle はオイラー角として、ヘッド角、ピッチ角、バンク角の3つの角度を保持する。モデルが -Z 方向を向いていると考えると、ヘッド角は Y 軸を回転軸とした回転を、ピッチ角は X 軸を回転軸とした回転を、バンク角は Z 軸を回転軸とした回転角を表す。

5.2 光源とマテリアル

一般に FKME ツールキットだけではなく 3 次元グラフィックスでは、描画シーン内に設置した光源と、表示オブジェクトの色を決定し、その両方のマテリアル情報をもとに計算を行うことでライティングが行われる。

FKME ツールキットには、「fk_Light」と呼ばれる光源を司るクラス、「fk_Color」と呼ばれる RGB 色情報を管理するクラス、「fk_Material」と呼ばれるマテリアルの色情報を保持するクラスと、「fk_MaterialAdmin」とよばれるマテリアルを管理するクラスがある。光源は形状の一つとして扱われる。

光源には平行光源、点光源、スポットライトの3種類がある。平行光源はあらゆる面に同一方向から照らす光源である。点光源はある一点から光を放射する光源で、室内の灯りなどに使用される。スポットライトは点光源のうち、ある一定方向に照射する光源である。光源から出る光には環境光、拡散光、鏡面光の3種類があり、それぞれ環境光は物体に当たって散乱した光、拡散光は直接光源から発せられる光、鏡面光は直接光源

から発せられる光のうち、金属光沢などに関係する。

fk_Material は形状の色を表現するクラスである。光源から出る 3 種類の光の反射に対応した環境反射色、拡散反射色、鏡面反射色の 3 種類の色情報を持ち、それ以外に放射色、透明度がある。環境反射色は、環境光に対しての反射の度合いを表すもので、物体の色に関係なく環境光から反射される色の度合いを表す。拡散反射色は光に左右されない物体本来の色を表し、鏡面反射色は反射の色合いを表す。放射色は物体自体が発光している色を表す。また透明度は物体の透明な質感を表す。

5.3 形状表現

FKME ツールキットでは描画できる形状として、ポリライン、ポリゴン、球、直方体、VRML データ、矩形テクスチャ、光源を定義することができる。VRML データは前述した mfg 形式のものを描画できる。クラスサイズを制限するために、光源以外は全て「fk_Solid」と呼ばれるクラスに統一してある。

球と光源を除き、定義できる形状には独自に特殊なメソッドが用意されている。

ポリラインとポリゴンは頂点座標を与えないと何も描画されない。頂点座標を追加するには pushVertex ()メソッドを使用する。

直方体は縦、横、奥行きを指定できる。デフォルトでは全て「1」となっている。直接サイズを設定するメソッドとして setSize()、サイズの倍率を設定するメソッドとして setScale()が用意されている。

VRML データを形状として定義した fk_Solid には、VRML データを変換した mfg 形式のファイルを読み込む setFigure()メソッドが用意されている。また、同様にテクスチャデータを変換した mtX 形式のファイルを読み込むメソッド setTexture()が用意されており、それぞれ引数にはリソース内のファイルの URL を指定する。

矩形テクスチャは長方形のテクスチャを生成するもので、readBMP()メソッドにより画像を読み込む。サイズは setTextureSize()メソッドにより変更することができる。今現在 24bit の WindowsBMP 画像しかサポートしていない。

5.4 モデル

モデルは FKME ツールキットの中でも最も重要な役割を果たす部分で、「fk_Model」というクラスで実現される。モデルは形状データ、位置ベクトル、アップベクトル、方

向ベクトル、オイラー角、マテリアル、モデル間の継承関係を保持する。位置や姿勢、方向から内部的に回転、移動の演算を行っているので、ユーザは複雑な行列演算などを意識することなく容易にモデルを制御することができる。形状データを登録するには、`setShape()`メソッドを使用する。また、位置座標やベクトルを取得することも可能で、他モデルから位置やベクトルを参照したいときなどに役立つと考えられる。これは後述する `glFocus()` などと組み合わせて用いられるときに力を発揮する。例えば、あるモデルが他のモデルを追尾するプログラムなどを非常に容易に記述することができる。

5.4.1 座標系

3次元コンテンツを作成する際に重要な概念に、グローバル座標系とローカル座標系がある。グローバル座標系はシーンの原点を中心とした座標系であり、モデルの位置に関わらず一定である。ローカル座標系は、主にモデルの継承に使う座標系で、モデルの位置を原点とし、方向ベクトルとアップベクトルをそれぞれ z 軸と y 軸とする座標系である。Z3D グラフィックスライブラリには、ローカル座標という概念がなく、全てグローバル座標をもとに形状の制御を行わなければならない。それは多くの場合、モデルを継承したときに開発者が子モデルの制御に悩まされることを意味する。FKME ツールキットではローカル座標系の概念を取り入れることにより、直感的にモデルの制御が可能となっている。ローカル座標系による制御を実現するために、モデルの位置と姿勢を制御するメソッドにグローバル座標系用とローカル座標系用のメソッドを用意している。表3にモデルに対する位置姿勢制御メソッドの一覧を示す。

表 3 : モデルに対する位置姿勢制御メソッド

制御名	メソッド名	必要な要素
グローバル平行移動	glTranslate	移動ベクトル
ローカル平行移動	loTranslate	移動ベクトル
位置指定移動	glMoveTo	位置ベクトル
グローバル注視点指定	glFocus	位置ベクトル
ローカル注視点指定	loFocus	位置ベクトル
グローバルオイラー角指定	glAngle	オイラー角
ローカルオイラー角指定	loAngle	オイラー角
方向ベクトル指定	glVec	方向ベクトル
アップベクトル指定	glUpVec	方向ベクトル
グローバル回転制御	glRotate	軸と角度
ローカル回転制御	loRotate	軸と角度
グローバル回転制御(方向固定)	glRotateWithVec	軸と角度
ローカル回転制御(方向固定)	loRotateWithVec	軸と角度

開発者はグローバル座標系メソッドとローカル座標系メソッドを使い分けることで自由に形状を制御することができる。注意すべき点は、モデル間に継承関係を持たせると、子モデルのグローバル座標系での位置と姿勢は親モデルのローカル座標系での位置と姿勢として扱われることである。

5.4.2 カメラの制御

FKME ツールキットの強力な機能の一つにカメラの制御がある。FKME ツールキットではどのようなモデルも、特別なクラスやメソッドを使うことなくシーンに登録するだけでカメラに設定することができ、モデルの注視点そのままカメラの注視点となる。これにより、Z3D グラフィックスライブラリでは非常に扱いにくいカメラの制御を、非常に簡便に行うことが可能となる。また、カメラはシーンに登録するカメラを入れ替えることで、簡単にカメラを切り替えることも可能である。これにより、モデルの視点やバードビューなどの切り替えを瞬時に行うことができる。

5.4.3 拡大率とマテリアル

FKME ツールキットでは他にモデルに対して、拡大率とマテリアルを設定することができる。直方体以外形状で大きさを変えることができないため、モデルの拡大率によって大きさを調節するとよい。また、マテリアルを設定することにより、モデルの色が決定される。

5.5 シーン

シーンは、一般的にディスプレイリストと呼ばれ、FKME ツールキットでは「fk_Scene」というクラスで定義される。fk_Scene はキャンバスに描画するモデルの一覧を登録しておく役割をもち、モデルを登録するには、entryModel()メソッドを使用する。また、カメラとしてのモデルを一つ登録しておく役割をもち、カメラを登録するには、entryCamera()メソッドを使用する。それぞれ引数には登録したいfk_Modelを指定する。

5.6 キャンバス

キャンバスは、モデルを描画するウィンドウであり、「fk_Canvas」というクラスで実現される。Z3D グラフィックスライブラリで定義されている、携帯上で3次元グラフィックスを実現するのに必要な C3DCanvas クラスを継承、拡張している。キャンバスの主な機能としては以下のものがある。

- ・ シーンの登録、保持
- ・ シーンから取得したモデルの描画
- ・ カメラの注視点によるビューイング変換処理
- ・ キーイベント処理

キーイベント処理は、デバイスの状態を取得するものであり、getKeyStatus()メソッドで与えられる。FKME ツールキットでは、キャンバスの内部の処理をカプセル化し、描画処理 (drawCanvas()メソッドとして与えられる) をループ内で呼び出すだけで、開発者

に内部の処理を意識させることなく描画やアニメーションを行うことができる。タイマーの定義とタイマーイベントの呼び出しという手順を踏まなくてもよいため、直接描画ループ内に値の変更を記述することができる。

5.7 実際の記述例

ここでは上述した FKME ツールキットのクラスライブラリを使用した記述例を示す。多少 FK システムとは異なるが、大筋は同じ書式を流用できる。

```
1: import com.nttdocomo.ui.*;
2:
3: public class Sample extends IApplication
4: {
5:     public void start(){
6:         fk_Canvas canvas = new fk_Canvas();
7:         fk_Solid block = new fk_Solid(FK.BLOCK);
8:         fk_Model blockModel = new fk_Model();
9:         fk_Model camera = new fk_Mode();
10:        fk_Model lightModel = new fk_Model();
11:        fk_Light light = new fk_Light();
12:        fk_Material lightMat = new fk_Material();
13:        fk_Scene scene = new fk_Scene();
14:
15:        blockModel.setShape(block);
16:        lightModel.setShape(light);
17:        lightMat.setAmbDiff(FK.fx1, FK.fx1, FK.fx1);
18:        lightModel.setMaterial(lightMat);
19:        scene.entryModel(blockModel);
20:        scene.entryModel(lightModel);
21:        scene.entryCamera(camera);
22:
```

```

23:     canvas.setScene(scene);
24:     Display.setCurrent(canvas);
25:     while(true){
26:         if(canvas.drawCanvas(true) == 0) break;
27:         blockModel.glRotateWithVec(0, 0, 0, FK.Y, FK.PI/60);
28:         if(canvas.getKeyStatus(FK.KEY_RIGHT) == true)
29:             camera.glTranslate(FK.fx1, 0, 0);
30:         if(canvas.getKeyStatus(FK.KEY_LEFT) == true)
31:             camera.glTranslate(-FK.fx1, 0, 0);
32:         if(canvas.getKeyStatus(FK.KEY_SELECT) == true)
33:             camera.glFocus(blockModel.getPosition());
34:     }
35: }
36: }

```

これは直方体が原点を中心に Y 軸を軸とした回転をするアニメーションと、左右ボタンを押すとカメラが左右に移動し、選択ボタンを押すとカメラが直方体のほうを向くという簡単なプログラムである。わずか 36 行でかけることに注目してもらいたい。改行を行わなければ 32 行である。このプログラムを、FKME ツールキットを使用しないで記述したとすると 150 行を超えてしまうだろう。

6 考察・検証

ここではツールキットに関する考察、課題、将来性について記述する。

6.1 ツールキットに関する考察

本研究に対する考察として、構築した FKME ツールキットを実装して、i アプリ上で 3 次元コンテンツを作成し、検証を行った。前章で示したとおり、FKME ツールキットを用いたプログラムはわずか 36 行で書けるのに対し、Z3D グラフィックスを用いて同

様のプログラムを作成した場合、150 行を超えた。これは十分、目的の一つであるプログラム行数の削減を達成しているといえるだろう。また、ループ処理型によりプログラムを作成する時間もかなり短縮化され、開発の効率化することができた。

形状に対する制御も FKME ツールキットを使うと非常に簡易に行うことができた。Z3D グラフィックスライブラリを使用して形状を制御しようとした場合、FKME で定義されているモデル制御メソッドの一つの制御を実現しようとするだけで、かなりの行数と数学的知識を必要とした。FKME ツールキットでは数学的知識をまったく必要とせず、様々な形状の制御をわずか 1 行で行うことができる。これは研究の目的の一つである強力かつ簡易な形状の制御を実現できたと考えられる。

以上のことから FKME ツールキットはツールキットとしての機能を十分に果たし、本研究の目的を達成することができたといえる。

6.2 ツールキットの課題

FKME ツールキットにより、i アプリにおける 3 次元コンテンツ開発の問題点を解消し、開発効率を向上させることができたが、いくつかの課題が生じてきた。課題について以下で述べる。

最も大きな課題は、ツールキットの肥大化である。必要な機能を実現した結果、ツールキット本体のサイズが JAR ファイル圧縮後で 23KB に達してしまった。これは残りのリソースと本体ソースコードの圧縮サイズを 7KB 以内に留めなければならないことを意味する。クラスサイズをさらに小さくすることも検討しているが、それでは必要な機能を削ることになりかねないので現段階では保留としている。アプリケーションの残り容量が 7KB だと、ソースコードとリソースを圧縮前で約 30KB 以内で作成しなければならない。ソースコードの行数でいうと 1200 行程度しか書くことができず、VRML モデルやテクスチャを使用するとさらに少なくなる。現状では FKME ツールキットを用いて 3 次元コンテンツを作成することはできるが、VRML モデルやテクスチャを多用したコンテンツを作成するのは難しい。

肥大化を抑えるためにツールキット本体のサイズを削減する工夫を凝らしているが、その結果、次のような問題が生じた。クラス構成の簡略化、メソッドの簡易化に伴い高度な形状生成や形状変形といった機能を除外したため、形状の生成をその種類ごとに、プリミティブに行う必要が生じた。この結果、FK システムの理念の一つである、高速

なりリアルタイム 3 次元グラフィックス描画に反することになってしまった。また、簡略化により、球や円などいくつかの形状が内部で生成できなくなった。さらに、メソッドの簡易化によるユーザビリティの低下も見られた。

固定小数点演算による誤差も大きな問題である。携帯電話では浮動小数点より表現できる数値の範囲が狭くなるが、高速な演算が可能である固定小数点しか扱えない。そのため FKME ツールキットでは固定小数点により演算を行っているが、移植元の FK システムでは浮動小数点演算による演算を行っている。浮動小数点演算を固定小数点演算に移植したため、内部の数値において多少の誤差が生じてしまう。また、携帯電話では角度を度数で計算しているが、FK システムでは弧度法であるラジアンを使用している。これは移植性の低下をもたらす可能性があるため、現在は円周率を与える定数を 180 とし、度数に変えることで移植性が落ちるのを防いでいる。

6.3 ツールキットの将来性

前述した問題点を解決する最短の解決法は、携帯電話本体のスペックアップである。現在の 30KB という i アプリサイズ制限ではやはりツールキットの性能を発揮することは難しい。実現したい機能があってもサイズ制限のため断念することもあった。以下に携帯電話本体がスペックアップし、サイズ制限が緩和されたことを前提として、FKME ツールキットの将来性について述べる。

使用可能なアプリケーションサイズの増加に伴い FKME ツールキットに、FK システムの様々な機能を追加することができる。現段階では形状の種類はポリライン、ポリゴン、直方体、球、VRML、矩形テクスチャの 6 つしか定義できないが、クラスサイズの制限が緩和されれば今回 FK システムから移植しなかった残りの形状を追加することができる。形状の生成・描画も、FKME ツールキットでは形状の生成・描画の両方を形状ごとに行っているが、形状の生成時に形状ごとに頂点座標を保持することにより、生成は形状ごと、描画は共通のメソッドを用いて一括して行うという形式をとることができ、これにより描画の高速化が図れる。また、多種類のテクスチャマッピングやデータベース、パーティクルなどの機能も実現できるようになるだろう。データベースの導入により動的な形状の変形などができるようになる。

今後 FKME ツールキットを Z3D グラフィックスエンジン専用から、マスコットカプセル用に移植していく予定である。マスコットカプセルの性能が Z3D グラフィックス

エンジンに追いついたと判断した時点で移植を開始する予定である。これによりまず D504i 専用ではなく、i アプリ対応の全機種に対応でき、汎用性を高めることができる。また、Java アプリと Ezplus にも対応できる可能性もある。将来的には機種も会社も関係ないマルチプラットフォームを実現していく必要があるだろう。

より機能を充実させ、マルチプラットフォームを実現することができたときに FKME は現在と比較してツールキットとしての真の意義を発揮できることになるであろう。

7 まとめ

現在、携帯電話上で Java アプリケーションを実行することができる Java 対応携帯電話は、全国に普及している携帯電話のうちおよそ 4 割を占め、多数の人が携帯上で 3 次元コンテンツを扱うことができるようになってきた。それに伴いハードウェアのほうでも三菱電機がハードウェアグラフィックスエンジンである Z3D グラフィックスエンジンを開発し、携帯電話用の高度な 3 次元コンテンツを開発することが可能となっている。3 次元コンテンツを開発する際に重要となる、複雑な操作や計算を簡易化してくれるツールキットはパーソナルコンピュータ用には存在するが、携帯電話用には 3 次元コンテンツを開発できるようになって間もないため、まだ存在しない。そこで本研究で携帯電話用のツールキットを作成することにした。FKME ツールキットは FK システムをベースとし、その理念を継承しており、リアルタイムにインタラクティブな操作を可能とし、簡易に携帯電話上で 3 次元コンテンツを開発することを可能としている。また、複雑で難解な形状の制御に必要な知識と作業をツールキット内部に格納してブラックボックス化することにより、開発者が意識することなく自由に形状を制御することを可能としている。FKME ツールキットは以上の機能を携帯電話用 Java プラットフォームで実現した。

FKME ツールキットを使用することでプログラム内のオブジェクトのつながりを、形状、モデル、シーン、キャンバスの 4 つのレイヤーを組み合わせることで明確にでき、わかりやすいプログラミングインタフェースを実現し、それにより 3 次元コンテンツの開発の際に問題となるプログラミング行数の多さも削減された。FKME ツールキットは携帯電話における 3 次元コンテンツの開発効率の向上を図ることができ、ツールキットとしての役割を携帯電話上において果たすことができた。

将来的には携帯電話のスペックアップとともに、FKME ツールキットの機能も拡張していく予定である。また、携帯電話の様々な機種に対応したマルチプラットフォームも実現していきたい。

謝辞

本論文を締めくくるにあたり、未熟な時分からご指導頂き本研究のきっかけ、適切な助言及び FKME ツールキットについてのご教授を頂きました、本校メディア学部の渡辺大地講師および和田篤氏（電気通信大学）に心より感謝申し上げます。

本研究において共同プロジェクトという形で、携帯電話におけるツールキットの在り方についてのヒントを頂いた本校メディア学部生の江原雅充氏に感謝します。

さらに本研究において、ご協力を頂いた三菱電機の皆様諸氏に感謝します。共同研究という形で色々ご教授頂きました。

本研究に御協力頂いたすべての皆様に、厚くお礼申し上げます。

参考文献・URL

- [1] BizMarketing, Java 対応携帯電話市場の動向, 2002/12/17,
<<http://www.bizmarketing.ne.jp/mob/020802293.shtml>>
- [2] Java Sun Micro System, 2002/12/17,
<<http://java.sun.com/>>
- [3] 株式会社 NTT ドコモ, 「i アプリコンテンツ開発ガイド～詳細編～」, 2002
<http://www.nttdocomo.co.jp/p_s/imode/java/>
- [4] 三菱電機, 「Z3D グラフィックスエンジン」, 2002/12/17
<<http://www.mitsubishielectric.co.jp/news/2001/1204-c.htm>>
- [5] エイチアイ、マスコットカプセル, 2002/12/17,
<<http://www.mascotcapsule.com/index-j.html>>
- [6] 三菱電機, 「D-Style」, 2002/12/17
<http://www.mitsubishielectric.co.jp/service/d-style/d_spec/index.html>
- [7] J. Neider, T. Davis and M. Woo, 「OpenGL Programming Guide」,
Addison-Wesley, 1992.
- [8] Microsoft, 「DirectX Home Page」, 2002/12/31
<<http://www.microsoft.com/japan/windows/directx/default.asp>>
- [9] 長島俊一、石田隆章、橘成泰, 「携帯電話・PHS 事業の現状」, 郵政研究所月報、
1999.7
- [10] 総務省, 「移動通信事業の動向」, 通信利用動向調査報告書事業所編、2001
- [11] J-PHONE, 「JAVA(TM)アプリ開発ガイド」, 2002
<<http://www.dp.j-phone.com/java/tech.php>>
- [12] au, 「Ezplus 技術情報」, 2002/12/25
<<http://www.au.kddi.com/ezfactory/tec/spec/ezplus.html>>
- [13] 三菱電機, 「ムーバ D504i」, 2002/12/17
<<http://www.mitsubishielectric.co.jp/mobile/mova/d504/index.html>>
- [14] 渡辺大地, 「リアルタイムグラフィックスのためのツールキットに関する研究」,
慶應義塾大学大学院政策・メディア研究科修士論文, 1996
- [15] 渡辺大地, 「Fine Kernel Tool Kit System」, 2002/12/17,
<<http://www.teu.ac.jp/media/earth/FK/>>

- [16] 大澤 範高, 「it3d(Interactive Toolkit library for 3D applications)」, 2002/12/17,
<<http://www.nime.ac.jp/it3d/>>
- [17] SGI, 「IRIS OpenGL Performer」, 2002/12/31
<<http://www.sgi.com/software/performer/>>
- [18] J.Rohlf and J.Helman, IRIS Performer, 「A High Performance Multiprocessing Toolkit for
Real-Time 3D Graphics, Computer Graphics」, 1995
- [19] 株式会社マックスネット, 「Open Inventor」, 2002/12/17,
<<http://www.maxnt.co.jp/MaxnetProducts/Tgs/oiv/overview/inventor.htm>>
- [20] J.Wernecke and Open Inventor Architecture Group, 「The Inventor Toolmaker :
Extending Open Inventor」, Release 2, Addison Wesley, 1994.