

Java を用いた Web3D のための 3 次元グラフィックスライブラリの開発

6 V - 0 1

陳 炳宇 西田 友是
東京大学

1. はじめに

この論文はインターネット上であるいは Web3D のために 3 次元のグラフィックスに対して新しいプログラミング環境について提案する。インターネット上で 3 次元のグラフィックスプログラムの開発は、OpenGL のような、普及した 3 次元のグラフィックスツールがないから容易ではない。そのため、1996 年の終わりからピュア Java を使って、我々は jGL と呼ばれる 3 次元のグラフィックスライブラリの開発を開始した [1]。jGL は汎用の 3 次元のグラフィックスライブラリであり、そのアプリケーション・プログラミング・インタフェース (API) は OpenGL の API と同様に定義される。また、OpenGL と同じ形式で記述できるので、プログラマーが簡単に使える。当時、マシンのパフォーマンスが良くなかったのでテクスチャマッピングのような複雑な計算が必要な機能を省略した。現在、ハードウェアはもっと良いが、ネットワークボトルネックはまだ前と比べて同じである。そこで我々は jGL の能力とパフォーマンスを高めるためにほとんどのコードを書き直し、インターネット上で伝送するためにコードサイズを最小にしたので報告する。

2. jGL の概要

jGL は Java バイナルマシン (VM) のために 3 次元のグラフィックスライブラリである。Java3D あるいは他のライブラリと異なり、jGL は前もってインストールされる必要がない (すべての必要なコードはランタイムにダウンロードされる)。jGL はピュア Java によって書かれているので、プラットフォームに依存せず、Java が使用可能などんなマシンの上でも実行させることができる。このほか、多くの 3 次元グラフィックスのプログラマーは OpenGL に慣れているので、jGL を使い易くするために、我々は OpenGL の仕様書 [2] に従って、jGL の API を OpenGL の API と同様に定義した。そうすることで、ユーザーがさらに他のライブラリを習う必要がなく、インターネット上で 3 次元グラフィ

ックスのアプリケーションを開発することが、以前より簡単になる。

jGL を用いてインターネット上で 3 次元グラフィックスのアプリケーションを開発する時、プログラマーは Java VM 中の 3 次元のグラフィックス・レンダリング機能を意識せずにすむ。すなわち、この 3 次元のグラフィックス・レンダリングは、jGL が行ってくれる。もし、この 3 次元プログラムを他のユーザーが使う場合、このプログラムと jGL を一緒にウェブサーバーに置いておく。そして、ユーザーはウェブブラウザのインラインアプレットでこのプログラムを使うことができる。

2.1 OpenGL vs. jGL

OpenGL の関数は 2 つの種類に分類することができる。図 1 (a) に示されるように OpenGL ユーティリティーライブラリ (GLU) と OpenGL (GL) である。jGL はこの図のように、関数の階層構造に従って開発した。

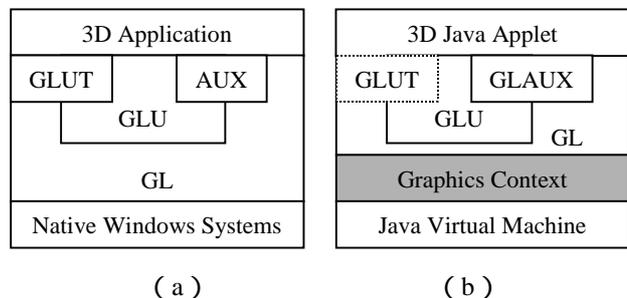


図 1: (a) OpenGL と (b) jGL の階層構造

2.2 パフォーマンスを高める課題

パフォーマンスは 3 次元のグラフィックスと Java アプリケーションの重要な課題である。そのため、jGL の大きな課題もパフォーマンスである。さらに、jGL はインターネットを介して操作するように設計されているため、インターネットのバンド幅も重要な問題と考えなければならない。そして、パフォーマンスを高める目標は 3 つある。jGL の能力を増加すること、パフォーマンスを高めること、および、コードサイズを最小にすることである。

そこで、以下に述べるような 4 つの点を考慮した。

「if-then-else」の構文を避けるために、クラス継承を使うこと、ルーチンの細分を試みること、分割したルーチンをグループ化すること、コー

The Development of a 3D Graphics Library for Web3D Platform by Using Java

Bing-Yu Chen and Tomoyuki Nishita
University of Tokyo

E-mail: {robin, nis}@is.s.u-tokyo.ac.jp

Web: <http://nis-lab.is.s.u-tokyo.ac.jp/~{robin, nis}/>

ドサイズを最小にするために、ファンクション・オーバー・ライディングの手法を使用することである。

3. 結果とまとめ

今回開発した jGL では 220 以上の OpenGL の関数を実装した。それには、2 次元および 3 次元の座標変換、3 次元の投影、デプス・バッファ、スムーズ・シェーディング、ライティング、材質特性、ディスプレイ・リスト、セクション、テクスチャ・マッピング、ミップ・マッピング、エバリュエータと NURBS が含まれている（アンチ・エイリアシングはサポートしていない）。なお、最近、GLUT を使う人が増えているので、この GLUT は追加予定である。

jGL の機能のテスト用に、26 個の例を我々の jGL のウェブページ¹で公開している。これらの例は、OpenGL の公式ガイドである *OpenGL Programming Guide* [3] から選んだものである。

jGL のパフォーマンスを評価するにあたり、図 2 のような、材質特性の異なる 12 個のボールをレンダリングするテスト・プログラムを使用した。なお、個々のボールは 256 ポリゴンでできている。このテスト・プログラムは *OpenGL Programming Guide* と同じ例である（Listing 6-3, pp. 183-184, Plate 16）。

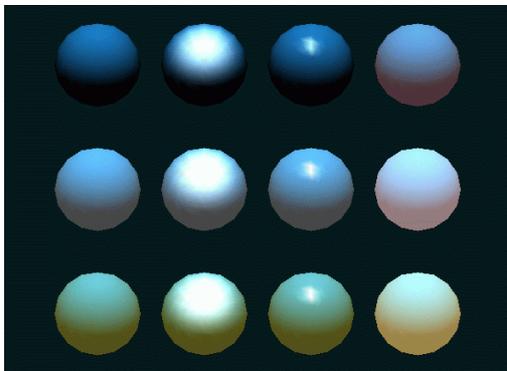


図 2 : jGL のパフォーマンスを評価する例。

Environment	Rendering Time (ms)	Platform
Sun JDK 1.3	219	Intel PentiumIII-1GHz, 512MB Microsoft Windows NT 4.0
	1061	Sun Ultra-10 360MHz, 256MB Sun Solaris 7

表 1 : jGL のパフォーマンスの評価

グラフィックス・エンジンとして用いる Java3D と比較するために、我々は Java3D のパッケージに HelloUniverse という例として色の異なるキューブを描くプログラムを書いた。結果は両方ともリア

ルタイムで表示できた。そのため、プログラムを使う前に、ユーザー側がランタイム・ライブラリをインストールする必要がないので、jGL は Web3D プラットフォーム上の小さいモデルに非常に適しているように思われる。

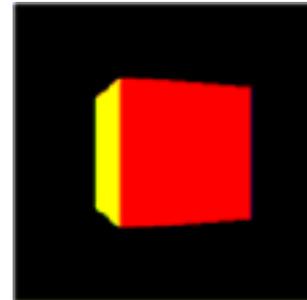


図 3 : jGL と Java3D のパフォーマンスの比較に用いた単純な例。

jGL の能力をテストするために、jGL を用いた VRML ブラウザ・アプレットを開発した。我々は VRML のスペック [4] に基づいてこのシステムの階層構造を決める。現在、VRML に関する約 70% ノードを実装した。また、ルートとイベント送信メカニズムを実装した。図 4はこのアプレットのスクリーン・ショットである。



図 4 : jGL を用いて VRML ブラウザ・アプレットでレンダリングする複雑な例。

参考文献

- [1] B.-Y. Chen, T.-J. Yang, and M. Ouhyoung, "JavaGL - a 3D Graphics Library in Java for Internet Browsers," in *IEEE Trans. on Consumer Electronics*, p.271 – p.278, Vol. 43, No. 3, 1997.
- [2] M. Segal, and K. Akeley, "The OpenGL Graphics Systems: A Specification (Version 1.1)," Silicon Graphics, Inc., 1996.
- [3] J. Neider, T. Davis, and M. Woo, "OpenGL Programming Guide," Addison-Wesley, 1993.
- [4] R. Carey, G. Bell, and C. Marrin, "ISO/IEC 14772-1:1997 Virtual Reality Modeling Language (VRML97)," The VRML Consortium Incorporated, 1997.

¹ <http://nis-lab.is.s.u-tokyo.ac.jp/~robin/jGL>