

# G3D ライブラリ

## 3D グラフィックスライブラリの概要

2008-05-30

任天堂株式会社発行

このドキュメントの内容は、機密情報であるため、**厳重な取り扱い、管理を行ってください。**

## 目次

1	はじめに .....	4
2	TWL-Systemの3D開発環境 .....	5
2.1	3D開発の流れ .....	5
2.2	バイナリコンバータ(g3dcvtr.exe) .....	5
2.3	G3Dバイナリファイル .....	6
2.4	プロジェクトにおけるリソース管理 .....	6
3	G3Dライブラリの概要 .....	7
3.1	G3Dライブラリの特長 .....	7
3.1.1	テキスト形式(XML)のファイルによるリソース管理が可能 .....	7
3.1.2	バイナリデータ .....	7
3.1.3	複数のCGツールで作られたデータの混在が可能 .....	7
3.1.4	NITRO-Viewerの出力がそのままゲーム画面に出力可能 .....	7
3.1.5	メモリのフラグメンテーションを起こしにくい .....	7
3.1.6	リソースへの名前によるアクセスが可能 .....	7
3.1.7	テクスチャ及びパレットの複数モデル間での共有が容易 .....	7
3.1.8	同一のアニメーションデータを異なる体型のモデルで再生可能 .....	8
3.1.9	モジュラリティの高さ、及びカスタマイズのしやすさ .....	8
3.2	G3Dライブラリの構成 .....	8

## 表

表 2-1	G3Dバイナリフォーマットの種類 .....	6
-------	------------------------	---

## 図

図 2-1	3D開発の流れ .....	5
図 3-1	G3Dライブラリの構成 .....	8

## 改訂履歷

改訂日	改訂内容
2008-05-30	NITRO-System の名称変更による修正 (NITRO-System を TWL-System に変更)。
2008-04-08	<ul style="list-style-type: none"> <li>改訂履歴の書式を変更。</li> <li>章の構成を変更。</li> <li>SOFTIMAGE 3D の記述を削除。</li> </ul>
2005-01-19	環境マップと投影マップについての記述を追加
2005-01-18	NITRO という表記をニンテンドーDS に統一。
2004-11-10	<ul style="list-style-type: none"> <li>概要と特長の説明を拡充。</li> <li>詳細についてはリファレンスマニュアルに集約。</li> </ul>
2004-09-16	<ul style="list-style-type: none"> <li>ima と ita をサポートした為、「3.1 G3D ライブラリの特徴」を修正。</li> <li>表 3.1 の誤記を修正。</li> <li>フラッグとフラグの二つの表記をフラッグに統一。</li> </ul>
2004-09-02	ノード数・マテリアル数・シェイプ数のリミットについての記述の追加。
2004-08-02	初版。

# 1 はじめに

G3D ライブラリは、ゲーム開発のために作成された 3D モデルやアニメーションを簡単かつ効率よく TWL やニンテンドーDS で表示、及び動作をさせることができるように開発された 3D 描画ライブラリです。G3D ライブラリには以下のような特長があります。

- 関節つきモデルの描画をサポートします。
- ジョイントアニメーション・マテリアルカラーアニメーション・テクスチャパターンアニメーション・テクスチャSRTアニメーション・ビジビリティアニメーションの再生をサポートします。
- 環境マップと投影マップをサポートします。
- Maya, SOFTIMAGE|XSI, 3ds max といった代表的な CG ツールで作成されたモデルの描画やアニメーションの再生に対応しています。
- NITRO-Viewer の 3D 描画エンジンは G3D を使用しているので、NITRO-Viewer で編集したキャラクタをそのままゲームに出すことができます。
- TWL およびニンテンドーDS 用に最適化がされていて、高速に動作します。
- モデルデータやアニメーションデータを XML 形式のテキストファイル(NITRO 中間ファイル)として管理することが可能です。
- ジョイント・マテリアル・テクスチャ・パレットにユーザーがつけた名前でアクセスすることが可能です。
- モーションブレンディングや部分的なジョイントアニメーションに対応しています。
- 複数のモデルデータの間でテクスチャやパレットを共有することが容易です。
- 形が異なってもジョイント構造が同じならば、同一のジョイントアニメーションを再生することが可能です。
- 明示的でない動的メモリ確保や解放を行わないので、フラグメンテーションや予期しないパフォーマンスの低下等のリスクを避けることができます。
- モジュール化されており、G3D ライブラリの動作をユーザー自身が細かくカスタマイズすることも可能です。

本ドキュメントでは、G3D ライブラリの概要について説明しています。G3D ライブラリや 3D バイナリコンバータの詳細については、リリースノートやオンラインリファレンスマニュアル等をご参照ください。

## 2 TWL-Systemの3D開発環境

### 2.1 3D開発の流れ

TWL-Systemの3D開発の流れは、図 2-1 に示したような形となります。デザイナーは、3Dモデリングツールで3Dモデルとアニメーションのデザインを行い、プラグインを使用してNITRO中間ファイルと呼ばれるXMLファイルを出力します。

3Dモデリングツールでは編集する事が出来ない TWL およびニンテンドーDS 固有のマテリアルの編集は、3Dマテリアルエディタを用いて行います。3Dマテリアルエディタでは、NITRO 中間ファイルのモデルデータ内のマテリアル部分を編集し、出力する事ができます。

プログラマは、3Dバイナリコンバータを用いて、NITRO 中間ファイルをG3Dライブラリで使えるバイナリファイルに変換し、アプリケーションで使用します。

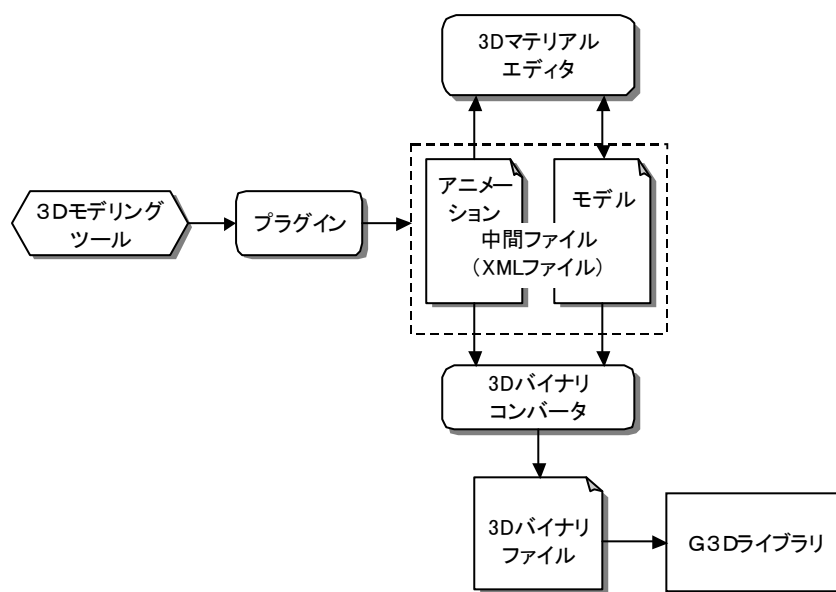


図 2-1 3D開発の流れ

### 2.2 バイナリコンバータ(g3dcvtr.exe)

バイナリコンバータ g3dcvtr.exe は、XML 形式（テキスト形式）である NITRO 中間ファイルをG3Dライブラリで用いるバイナリデータに変換するための Windows アプリケーションです。g3dcvtr.exe は NITRO 中間ファイルの全てを変換する事ができます。

バイナリコンバータ g3dcvtr.exe の詳細については、g3dcvtr のマニュアルをご参照ください。

(TwlSystem¥docs¥G3D¥g3dcvtr\_UsesManual.pdf)

## 2.3 G3Dバイナリファイル

G3Dバイナリファイルは、NITRO 中間ファイルからバイナリコンバータ `g3dcvtr.exe` により変換されます。このG3Dバイナリファイルの種類は、NITRO 中間ファイルの種類と対応しており、以下の6種類のフォーマットがあります。

表 2-1 G3Dバイナリフォーマットの種類

拡張子	中間ファイルの種類	ファイル内容
nsbmd	モデルデータ(imd)	ポリゴン、親子構造、マテリアル、テクスチャを含む モデルデータ
nsbtx	テクスチャデータ(imd)	テクスチャだけのデータ
nsbca	キャラクタアニメーションデータ(ica)	ノードの行列を操作するアニメーションデータ
nsbva	ビジュビリティアニメーションデータ(iva)	ノードのビジュビリティを操作するアニメーションデータ
nsbma	マテリアルカラーアニメーションデータ(ima)	マテリアルの色を操作するアニメーションデータ
nsbtp	テクスチャパターンアニメーションデータ(itp)	複数のテクスチャを差し替えるアニメーションデータ
nsbta	テクスチャSRTアニメーションデータ(ita)	テクスチャ行列を操作するアニメーションデータ

## 2.4 プロジェクトにおけるリソース管理

リソースファイルについては、プロジェクトツリー内に別途コンバートされたバイナリリソースファイルを置くのではなく、プロジェクトのビルドツリー内に NITRO 中間ファイルを置き、プログラムのビルド時にコンバートすることをお奨めいたします。理由は、異なるリリースのバイナリコンバータで変換されたバイナリでの動作はサポートされないからです。サポートされない理由としては、

- 過去のバージョンのバイナリをサポートすることによって、ライブラリのコードサイズやバイナリサイズが必要以上に増加してしまうこと。
- 必要以上にサイズの大きいライブラリはゲームプログラムにとって負担となり、ゲームの制作期間や制作コストに悪影響を与えてしまうこと。

等を挙げることができます。

過去のバイナリが動作する場合でも、新しいバージョンの `g3dcvtr` によってデータ変換の最適化が改善されることによって新しくコンバートしたバイナリの方が高速に動作するといったこともあります。

## 3 G3Dライブラリの概要

### 3.1 G3Dライブラリの特長

---

G3D ライブラリを用いることによって、TWL およびニンテンドーDS のハードウェア性能を引き出しつつ 3D ゲーム開発を効率化することができます。この節ではゲーム開発に G3D ライブラリを用いることのメリットについて簡単に解説をします。

#### 3.1.1 テキスト形式(XML)のファイルによるリソース管理が可能

---

モデルデータやアニメーションリソースはテキストファイル(NITRO 中間ファイル)形式で管理することができます。XML 形式のテキストファイルなので管理が容易です。

#### 3.1.2 バイナリデータ

---

NITRO 中間ファイルは 3D バイナリコンバータ(g3dcvtr)によってバイナリデータファイルに変換されます。G3D ライブラリは、DS カード上のバイナリデータファイルを読み込んで再生することができます。C のソースコードに変換する方式と比較すると実行時におけるモデルの入れ替え等が容易だというメリットがあります。

#### 3.1.3 複数のCGツールで作られたデータの混在が可能

---

G3D は複数の CG ツールで作成されたデータに対応します。1つのプログラム内で複数の CG ツールから作成されたデータを混在させることも可能です。ですから、それぞれの CG ツールの強みを生かしたゲーム開発が可能です。

#### 3.1.4 NITRO-Viewerの出力がそのままゲーム画面に出力可能

---

NITRO-Viewer は G3D を使用して 3D モデルの描画を行っているので、NITRO-Viewer で編集したキャラクタをそのままゲームに出すことができます。

#### 3.1.5 メモリのフラグメンテーションを起こしにくい

---

G3D ライブラリは API 内で暗黙的な動的メモリ確保や解放を行いません。ロードしたバイナリデータはそのままの形で描画に利用できるため、初期化においても動的なメモリ確保を行うことはありません。更に複数のモデルデータやアニメーションデータを g3dcvtr により 1 つのバイナリファイルにまとめることが可能です。このことにより、G3D を使用するプロジェクトはフラグメンテーションやパフォーマンス低下の危険を低減させることができます。

#### 3.1.6 リソースへの名前によるアクセスが可能

---

ジョイント・マテリアル・テクスチャ・パレットはユーザーがリソース作成時につけた 16 文字以内の名前でアクセスすることが可能です。このことにより、プログラムコードのデータへの依存度を減らすことができます。

#### 3.1.7 テクスチャ及びパレットの複数モデル間での共有が容易

---

3D バイナリコンバータ(g3dcvtr)で複数のモデルを 1 つのバイナリファイルにコンバートすることができます。この場合、テクスチャやパレットで同一のものがある場合はデータが共有されるようになります。また、モデルとテクスチャを別個のバイナリファイルにコンバートして実行時に関連付けるといったことも可能です。

### 3.1.8 同一のアニメーションデータを異なる体型のモデルで再生可能

ジョイント構造が同一であるならば、大きさや体型が異なるモデル間で同一のアニメーションデータを共有することが可能です。この機能を利用すれば、アニメーションデータ作成作業の手間や、メモリ使用量を節約することができます。

### 3.1.9 モジュールリティの高さ、及びカスタマイズのしやすさ

G3D ライブラリのアニメーションの計算・アニメーションのブレンド・個々CG ツール対応コードはモジュール化されており、追加・削除することが容易です。使用するモジュールを動的に切り替えることも可能です。

## 3.2 G3Dライブラリの構成

G3Dライブラリは、図 3-1 に示すような構成となっています。

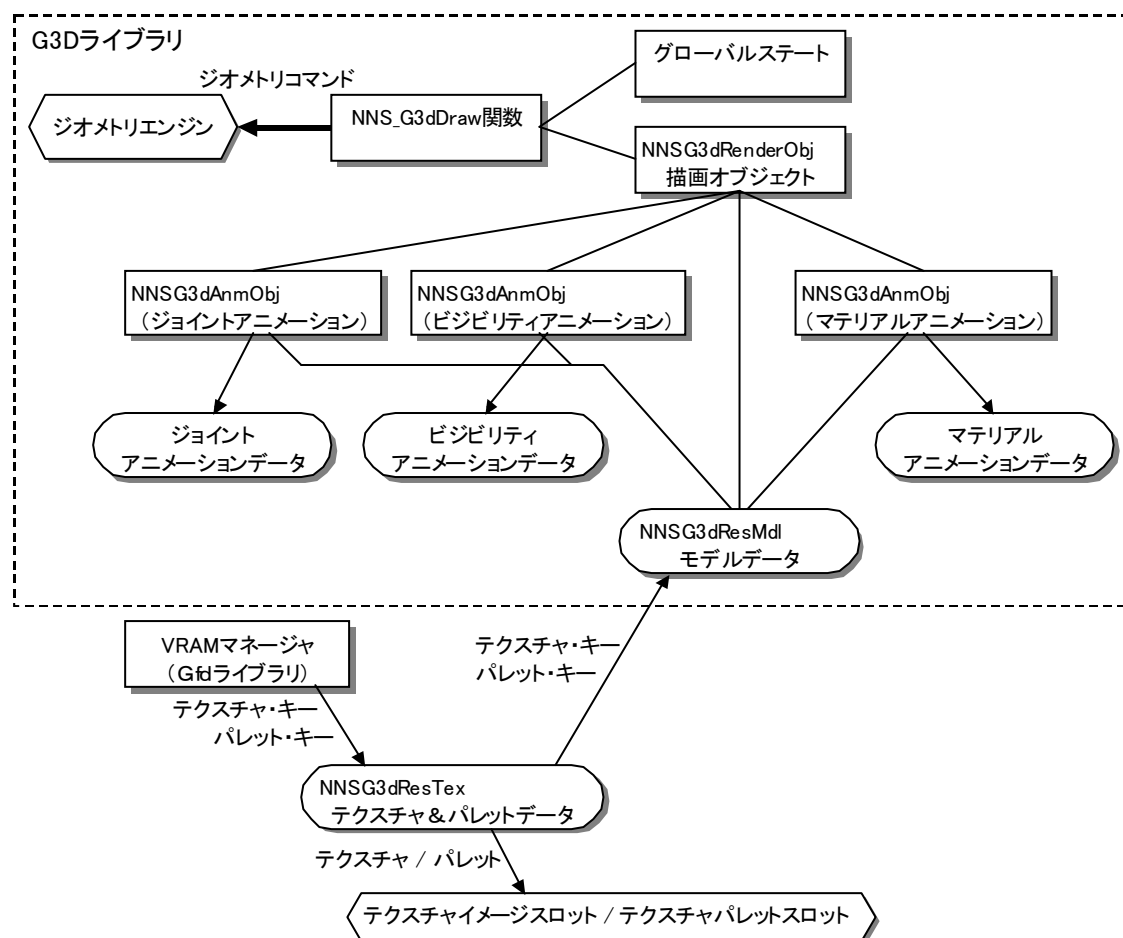


図 3-1 G3D ライブラリの構成



Windows は米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Softimage、SOFTIMAGE|XSI は米国 Avid Technology,Inc. の登録商標または商標です。

3ds max、Maya は Autodesk,Inc./Autodesk Canada,Inc. の米国およびその他の国における登録商標または商標です。

その他、記載されている会社名、製品名等は、各社の登録商標または商標です。

© 2004-2008 Nintendo

任天堂株式会社の許諾を得ることなく、本書に記載されている内容の一部あるいは全部を無断で複製・複写・転写・頒布・貸与することを禁じます。