

TWL-System

G3D Library

Overview of 3D Graphics Library

2008/05/30

**The content of this document is highly confidential
and should be handled accordingly.**

Confidential

These coded instructions, statements, and computer programs contain proprietary information of Nintendo and/or its licensed developers and are protected by national and international copyright laws. They may not be disclosed to third parties or copied or duplicated in any form, in whole or in part, without the prior written consent of Nintendo.

Table of Contents

1	Introduction	5
2	TWL-System 3D Development Environment	6
2.1	Flow of 3D Development.....	6
2.2	Binary Converter: g3dcvtr.exe	6
2.3	G3D Binary Files.....	7
2.4	Resource Management in the Project	7
3	Overview of the G3D Library	8
3.1	Characteristics of the G3D Library.....	8
3.1.1	Manage Resources with a Text (XML) File	8
3.1.2	Read Binary Data from a DS Card	8
3.1.3	Mix Data Sets Created with Different CG Tools	8
3.1.4	Output from NITRO-Viewer to the Game Screen.....	8
3.1.5	Less Likely to Cause Memory Fragmentation.....	8
3.1.6	Access Resources by Name	8
3.1.7	Textures and Models in the Palette Can be Shared Easily	9
3.1.8	Play Back the Same Animation Data with Models of Different Shapes	9
3.1.9	High Modularity and Easy Customization	9
3.2	Organization of the G3D Library	9

Tables

Table 2-1 G3D Binary Format Types	7
---	---

Figures

Figure 2-1 Flow of G3D Development	6
Figure 3-1 G3D Library Organization.....	9

Revision History

Date	Description
2008/05/30	Made revisions in line with the NITRO-System name change (from NITRO-System to TWL-System).
2008/04/08	<ul style="list-style-type: none">• Changed the format of the Revision History.• Changed the section structure.• Deleted the description of SOFTIMAGE 3D
2005/01/19	Added descriptions of environmental and projection mapping.
2005/01/18	Changed some instances of "NITRO" to "Nintendo DS."
2004/11/10	<ul style="list-style-type: none">• Improved and expanded the overview and introduction.• Put all the details in the reference manual.
2004/09/16	<ul style="list-style-type: none">• Modified "Special Characteristics of G3D Library" to support ima and ita.• Corrected Table 3-1.• Unified the spelling of "flag" in the Japanese edition.
2004/09/02	Added description on limiting the number of nodes, materials, and shapes.
2004/08/02	Initial version.

1 Introduction

The G3D Library is a 3D graphics library developed to allow simple and effective display and movement of 3D models and animation created for game development on a TWL or Nintendo DS system.

G3D library provides the following features:

- Supports drawing of models that have joints.
- Supports playback of joint animation, material color animation, texture pattern animation, texture SRT animation, and visibility animation.
- Supports environmental mapping and projection mapping.
- Supports drawing models and playing animations that were created with common CG tools such as Maya, SOFTIMAGE |XSI, and 3ds Max.
- Outputs characters edited with NITRO-Viewer to the game as is because the NITRO-Viewer's 3D drawing engine uses G3D.
- Works fast because it is optimized for TWL and Nintendo DS systems.
- Manages model and animation data as XML-format text files (NITRO intermediate files).
- Uses names specified by the user to access joints, materials, and texture palettes.
- Supports motion blending and partial joint animation.
- Simplifies sharing textures and palettes between multiple sets of model data.
- Allows you to play the same joint animation with different shapes as long as the joint structure is the same.
- Avoids risks such as fragmentation or unexpected performance reduction because it does not allocate or release the dynamic memory that is not specified explicitly.
- Allows the user to customize the operation of the G3D library because it is provided in separate modules.

This document explains the operation of the G3D library. For details on G3D library and 3D binary converter, see the release notes and Function Reference manual.

2 TWL-System 3D Development Environment

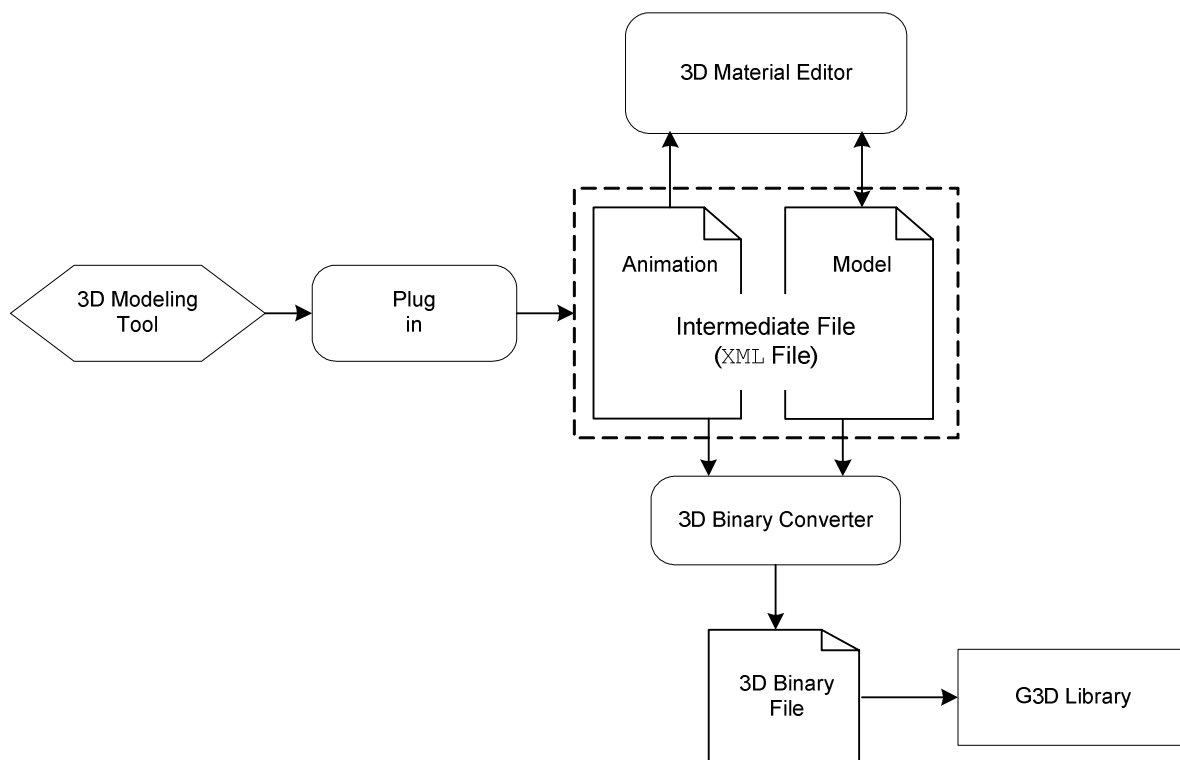
2.1 Flow of 3D Development

The flow of 3D development for the TWL-System is shown in Figure 2-1, below. The designer creates a 3D model and animation using a 3D modeling tool, and then uses a plug-in to output it as an XML file. The XML file is a *NITRO intermediate file*.

Specific materials for TWL and Nintendo DS systems that cannot be edited with the 3D-modeling tool can be edited with the 3D Material Editor. With the 3D Material Editor, the material portion of the model data in the NITRO intermediate file can be edited and output.

Using the 3D binary converter, the programmer converts the NITRO intermediate file to a binary file format. The binary file can be used in the application via the G3D Library.

Figure 2-1 Flow of G3D Development



2.2 Binary Converter: g3dcvtr.exe

The binary converter (`g3dcvtr.exe`) is a Windows application to convert an XML-format NITRO intermediate file to binary data used by the G3D Library. `g3dcvtr.exe` can convert all types of NITRO intermediate files. For detailed information about `g3dcvtr.exe`, see the `g3dcvtr` manual (`NitroSystem\docs\G3D\g3dcvtr_UsersManual.pdf`).

2.3 G3D Binary Files

The G3D binary file is converted from the NITRO intermediate file by `g3dcvtr.exe`. The G3D binary file types correspond to the NITRO intermediate file types. The six file types are shown below.

Table 2-1 G3D Binary Format Types

Extension	Intermediate File Type	File Description
nsbmd	Model data (imd)	Model data including polygons, parent/child structure, materials, and textures.
nsbtx	Texture data (imd)	Texture-only data.
nsbca	Character animation data (ica)	Animation data that operates the node matrix.
nsbva	Visibility animation data (iva)	Animation data that operates the node visibility.
nsbma	Material color animation data (ima)	Animation data that operates the material colors.
nsbtp	Texture pattern animation data (itp)	Animation data that cycles through multiple textures.
nsbta	Texture SRT animation data (ita)	Animation data that operates the texture matrix.

2.4 Resource Management in the Project

Instead of individually placing converted binary resource files in the project tree, we recommended that you place the NITRO intermediate files in the project's build tree and convert the files during the program build. This is because the operation of binary files converted with different releases of the binary converter is not supported. Different releases of the converter are not supported because:

- Including support for older versions of binary files would make the library code and binary sizes much larger than necessary.
- Oversized libraries can create increased burden on game programs and have detrimental effects on game production in terms of both time and cost.

Even though previous binary files may work, newly converted files work faster because of data-conversion improvements in `g3dcvtr`.

3 Overview of the G3D Library

3.1 Characteristics of the G3D Library

The G3D library can take advantage of TWL and Nintendo DS hardware features to make 3D game development more efficient. This section explains the advantages of using the G3D Library for game development.

3.1.1 Manage Resources with a Text (XML) File

Model data and animation resources are managed by NITRO intermediate files. The XML files are in text format, which is easy to understand.

3.1.2 Read Binary Data from a DS Card

A NITRO intermediate file is converted to a binary data file by the 3D binary converter (g3dcvtr). The G3D library can load binary data files from a DS Card and play the file. Compared to the method that converts to C source code, models can be replaced more easily during execution.

3.1.3 Mix Data Sets Created with Different CG Tools

The G3D library can handle data created with different CG tools, allowing you to mix data sets created with different CG tools in a single program. Therefore, it is possible to develop games that get the most out of each CG tool.

3.1.4 Output from NITRO-Viewer to the Game Screen

NITRO-Viewer uses G3D to draw 3D models, so the characters edited with NITRO-Viewer can be output to the game as is.

3.1.5 Less Likely to Cause Memory Fragmentation

The G3D library does not perform implicit allocation or release of dynamic memory in the API. Loaded binary data can be used for drawing without any changes, so it does not perform dynamic memory allocation even during the initialization. Additionally, it is possible to collect multiple sets of model and animation data into one binary file by using g3dcvtr. Consolidating the data into a single file that uses the G3D library reduces the probability of memory fragmentation and performance reduction.

3.1.6 Access Resources by Name

Joints, materials, and texture palettes can be accessed by the name specified when the resource was created. (The user-specified name must be 16 characters or less.) This decreases the degree of dependency on the data in the program code.

3.1.7 Textures and Models in the Palette Can be Shared Easily

The 3D binary converter can convert multiple models into one binary file, in which identical textures and models in the palette share data. Also, models and texture can be converted into different binary files and associated during execution.

3.1.8 Play Back the Same Animation Data with Models of Different Shapes

As long as the joint structure is the same, the same animation data can be shared among the models with different sizes or shapes. This saves you time you would otherwise use to re-create animation data and reduces the amount of memory usage.

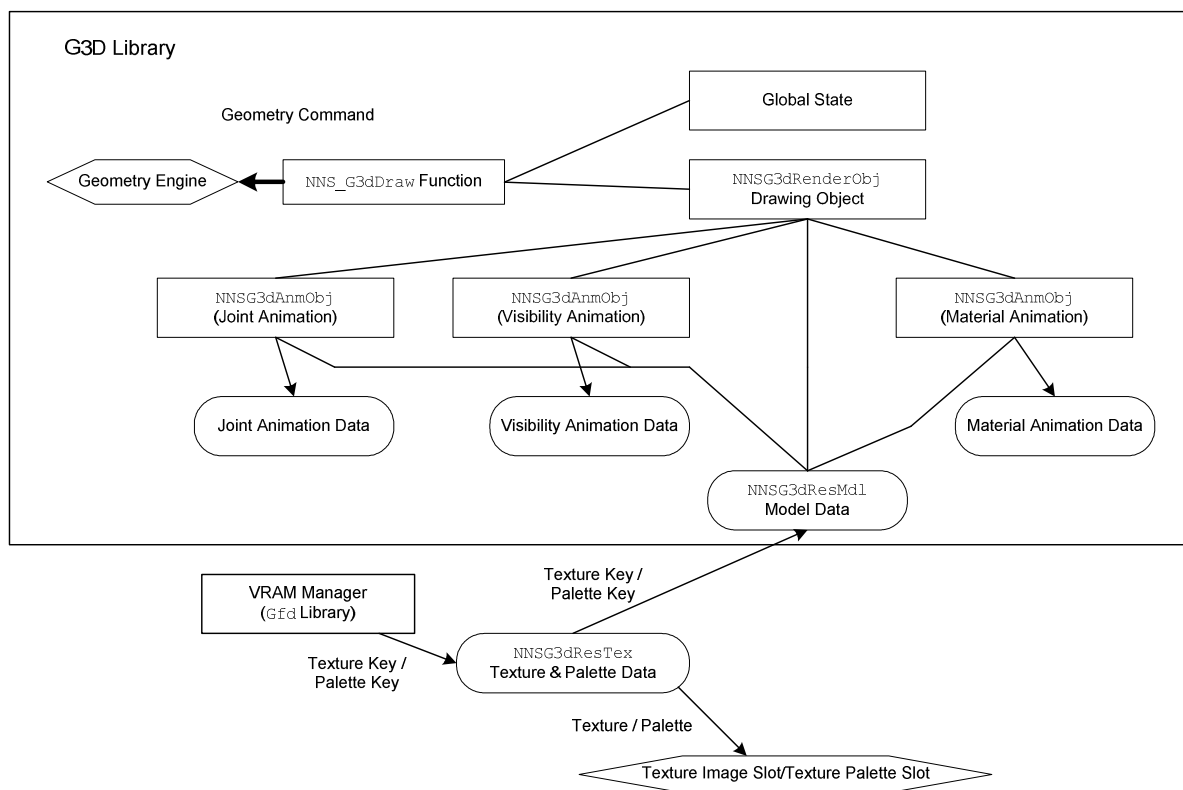
3.1.9 High Modularity and Easy Customization

Animation calculation, animation blending, and codes that support individual CG tools are made into modules that can be added or removed easily. The modules can be switched dynamically.

3.2 Organization of the G3D Library

The G3D Library is organized as shown in Figure 3-1.

Figure 3-1 G3D Library Organization



Windows is a registered trademark or trademark of the Microsoft Corporation in the U.S. and in other countries.

Softimage and SOFTIMAGE|XSI are registered trademarks or trademarks of Avid Technology, Inc. in the U.S.

3ds Max and Maya are registered trademarks or trademarks of Autodesk, Inc./Autodesk Canada, Inc, in the U.S. and other countries.

All company and product names in this document are the trademarks or registered trademarks of the respective companies.

© 2004-2009 Nintendo

The contents of this document cannot be duplicated, copied, reprinted, transferred, distributed, or loaned in whole or in part without the prior approval of Nintendo.