TWL-System G2D Library

2D Graphics Overview

2008/05/30

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

Table of Contents

Tables

Figures

Revision History

Revision date	Description of revisions
2008/05/30	Made revisions in line with the NITRO-System name change (from NITRO-System to TWL-System).
2008/04/12	Changed the format of the cover page and the Revision History.
2007/03/14	Revised the description of the file format.
2005/01/05	Changed several instances of “NITRO” to “Nintendo DS”.
2004/12/06	Standardized phrases. Overall revision. Moved the description of details to API reference.
2004/09/16	Added BG functionality.
2004/08/02	Added support for August 2 version. Corrected typos.
2004/07/20	Added support for July 20 version.
2004/06/22	Added support for June 22 version.
2004/06/10	Added support for June 10 version. Corrected typos.
2004/05/28	Added support for May 28 version.
2004/05/12	Correction of typos. Correction of exhibits.
2004/05/10	Initial version.

1 Introduction

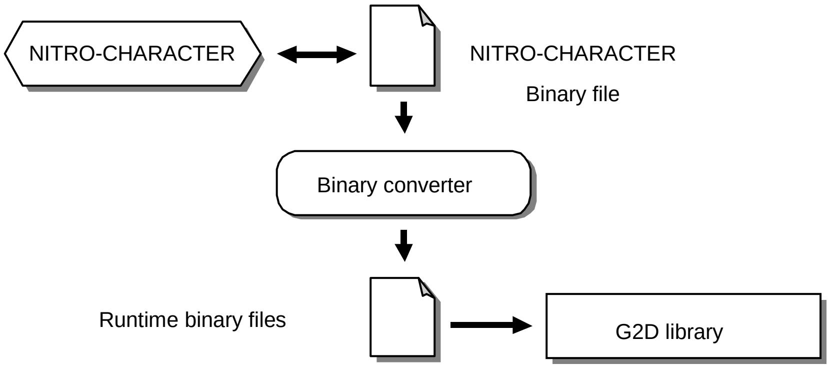
Both the TWL and Nintendo DS can draw OBJs and backgrounds with the 2D graphics engine and polygons with the 3D graphics engine. To draw more sophisticated 2D representations, the TWL-System 2D graphics library can use both the 2D and the 3D graphics engines.

2 TWL-System 2D Development Environment

2.1 2D Development Flow

The 2D development flow for TWL-System is shown in Figure 2-1 2D Development Environment, below**Error! Reference source not found.**. To design characters and OBJ animations, graphic designers use a 2D design tool called NITRO-CHARACTER. To convert the NITRO-CHARACTER binary files, programmers use the binary converter. The binary files are saved by NITRO-CHARACTER to the run-time binary files, which can be used by the G2D library (a part of the 2D graphics library).

Figure 2-1 2D Development Environment



2.2 NITRO-CHARACTER

NITRO-CHARACTER is a Windows application that creates the character data used for drawing and editing dot images, screen data, and OBJ animation data. These data are used to create 2D screens for the TWL and Nintendo DS.

2.3 NITRO-CHARACTER Binary Files

NITRO-CHARACTER saves data in the NITRO-CHARACTER binary files. Among various types of other data, these files contain the data specific to NITRO-CHARACTER and those used by the TWL and Nintendo DS 2D graphics engine. The files can be loaded back into NITRO-CHARACTER to restore the work. Table 2-1 shows the seven types of NITRO-CHARACTER binary files.

Table 2-1 NITRO-CHARACTER Binary Format Types

File type	Extension	File Extension Explanation
Scene files	`.nsn`	**N**itro **S**ce**N**e.
Color palette files	`.ncl`	**N**itro **C**o**L**or Palette.
Character files	`.ncg`	**N**itro **C**haracter **G**raphics.
Cell files	`.nce`	**N**itro **Ce**ll.
Multicell files	`.nmc`	**N**itro **M**ulti**C**ell.
Screen files	`.nsc`	**N**itro **Sc**reen.
Preview files	`.npv`	**N**itro **P**re**V**iew.

2.4 G2D Runtime Binary Format

G2D runtime binary format is the data format that can be processed by the G2D library. Mainly, these data are created using the binary converter described in paragraph 2.5 Binary Converter. Table 2-2 shows the nine types of G2D runtime binary formats.

Table 2-2 G2D Runtime Binary Format Types

File type	Extension	File Extension Explanation
Color palette files	`.nclr`	**N**itro **C**o**L**or Palette for **Run-Time**.
Character files	`.ncgr`	**N**itro **C**haracter **Gr**aphics for **Run-Time**.
Character files (bitmaps)	`.ncbr`	**N**itro **C**haracter **B**itmap Format for **Run-Time**.
Cell files	`.ncer`	**N**itro **CE**ll for **Run-Time**.
Multicell files	`.nmcr`	**N**itro **M**ulti**C**ell for **Run-Time**.
Cell animation files	`.nanr`	**N**itro Cell **AN**imation for **Run-Time**.
Screen files	`.nscr`	**N**itro **SC**reen for **Run-Time**.
Multicell animation files	`.nmar`	**N**itro **M**ulticell **A**nimation for **Run-Time**.
Entity files	`.nenr`	**N**itro **EN**tity for **Run-Time**.

2.5 Binary Converter

The binary converter `g2dcvtr.exe` is a Windows application that converts the binary files generated by NITRO-CHARACTER into the runtime binary files, a format that can be used by the G2D library. For details about the binary converter and the runtime binary data format, see the *g2dcvtr Manual* (`TwlSystem\docs\G2D\g2dcvtr_Manual.pdf`) and *G2D Library Runtime Binary Format* (`TwlSystem\docs\G2D\NNS_G2DRuntime-BinaryFormat.pdf`).

2.6 G2D Library

The G2D library allows the user to easily generate 2D graphics for the TWL and Nintendo DS. When used with NITRO-CHARACTER (see section 2.2 NITRO-CHARACTER), G2D provides the functionality needed to create sophisticated graphics on the TWL and Nintendo DS system.

3 Overview of the G2D Library

3.1 Library Functionality

This section describes the functionality and capabilities of the G2D library.

3.1.1 Playback of NITRO-CHARACTER Animation Data

Using the animation data output by NITRO-CHARACTER allows you to perform pattern animation of game characters.

3.1.1.1 Controlling Animation Playback

The playback methods such as playback, reverse playback, slow playback, and loop playback can be changed in the program. This feature supports various playback modes that can be set with NITRO-CHARACTER.

3.1.1.2 Calling Callback Functions at Various Times within the Animation

You can call callback functions at various times during animation playback. It is also possible to register 4-byte user parameters with callback functions.

Related modules:

Cell animation, animation controller

3.1.2 OAM Management

The OAM management feature gives you detailed control of the 2D graphics engine’s OAM.

3.1.2.1 Manage the Registration of OAM Attribute and Affine Parameter

You can specify the area managed by the OAM manager.

3.1.2.2 Draw Software Sprite of the Registered OAM Attribute

Using 3D polygons, this function draws the OBJ registered in the OAM manager without registering the OBJ to OAM.

3.1.2.3 Split Display When There Are More OBJs than the OAM Attributes Being Used (Extended OAM Manager)

This feature splits the display when the number of OAM attributes being used is larger than the number of OBJs. This feature blinks while it is being displayed.

Related modules:

OAM manager, extended OAM manager

3.1.3 Drawing Sprites Using the 3D Graphics Engine

The G2D library provides a feature called a software sprite that uses square polygons to draw sprites with the TWL and Nintendo DS 3D graphics engine. Compared with the 2D graphics engine, a software sprite gives you a higher degree of freedom to draw an object.

G2D also provides the OAM software sprite drawing module that interprets the contents of the OAM attribute and provides the API that draws software sprite. With this module, you can draw the same OBJ data with software sprite or (hardware) OBJ. Due to hardware restrictions, both the character data for software sprite and the character data for OBJ must be stored in VRAM when switching between software sprite and OBJ. Character data for software sprite and for OBJ are in different formats, but the feature that converts them is provided when the converter is used.

The following subsections describe the features of software sprites.

3.1.3.1 Number of Sprites

The number of sprites that can be displayed with software sprites is larger than the number that can be displayed using OBJ and the 2D graphics engine. Because the TWL and Nintendo DS display each software sprite on a quadrilateral polygon, the total number of software sprites is limited by the number of quads that the TWL and Nintendo DS can draw. If polygons are not used for any other purpose, the TWL and Nintendo DS can display 1536 software sprites.

3.1.3.2 Sprite Rotation and Scaling

Software sprites have the limitation on the number of affine parameters in the OBJ functionality of the 2D graphics engine. Also, with this functionality characters cannot be expanded to more than double their size, but sprites can be displayed properly at larger sizes.

3.1.3.3 Translucency Feature

The transparency level of software sprites ranges from 0 (zero) to 31. If the transparency level is set to 31, the sprite is opaque. If the transparency level is set to 0 (zero), the sprite is transparent and does not appear on the screen.

Related modules:

Software sprite, OAM software sprite drawing

3.1.4 Loading Image Data to VRAM

The G2D library provides a feature for loading the image data to VRAM. It also provides the mechanisms, such as image proxy and image palette proxy, for managing loaded image data.

Related modules:

Load Image and others

3.1.5 Complex Animation Composed of Pattern Animations

With NITRO-CHARACTER, one large game character can be created by combining small components which are pattern-animated using different frame counts. This is called a multicell animation. In addition, you can build complex game characters by animating the various components of a character at different speeds.

The G2D library supports multicell data drawing and animation.

Related modules:

Multicell

3.1.6 VRAM Transfer Animation

The G2D library provides a system called VRAM Transfer Animation, which allows you to render animation by transferring the image (VRAM) data referenced by the cell and rewriting the data.

Using the VRAM transfer animation feature, you can create animations with a large amount of image data while reducing the image data capacity required by the hardware. In 2D game development, the VRAM transfer animation is generally used for developing a main character that has many animation patterns.

3.1.6.1 Avoid Unnecessary VRAM Transfer

The state of the VRAM transfer animation is managed by the cell VRAM transfer state object. The cell VRAM transfer state object stores the state information (for example, whether the animation switched or drawn). The library uses this information to do the VRAM transfer only as needed.

Related modules:

Cell animation, cell VRAM transfer state manager

3.1.7 Sophisticated Drawing with a High-Level Drawing Module

G2D library provides a high-level drawing module that allows sophisticated drawing.

3.1.7.1 Handling the Main Screen and Sub-Screen as One Contiguous Screen

The two screens of the TWL and Nintendo DS can be handled as one large, contiguous screen that can be placed anywhere in the space shared by the two screens.

3.1.7.2 Affine Transformation Parameter Management

This feature monitors the affine transformation and sets the affine parameter automatically. To reduce the number of affine parameters in use, as much as possible share those affine parameters that can be shared. Flip conversion of a game character that has been affine transformed is done by automatically calculating the affine matrix.

3.1.7.3 Dynamically Switching the Drawing of Software Sprites and OBJs

This feature allows you to dynamically switch the drawing methods.

3.1.7.4 Overwriting the OBJ Parameters for Drawing

By specifying a value, you can overwrite the OBJ parameter for drawing.

Related modules:

Renderer, Renderer core

3.1.8 BG Automatic Composition

From the NITRO-CHARACTER binary data, the G2D run-time screen data inherit the information required for the display of the BG. The G2D library uses these data to provide functionality for the automatic loading of BG control and graphics data into VRAM. With this feature, the programmer can perform common initialization tasks, allocate VRAM, and display the BG by enabling its display status.

Related modules:

BG