NintendoWare for CTR

Programmer's Guide

2010/07/29

Version 1.7

PROVISIONAL TRANSLATION

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

Table of Contents

Figures

1 Introduction

By using the runtime libraries included in NintendoWare for CTR (hereinafter "NWC4"), you can take graphics and sound data generated by NWC4 tools and display and play it back on a CTR system.

This document is targeted at programmers who use the NWC4 runtime libraries. It explains procedures for building sample programs, the structure of the runtime libraries, and so on.

See the Function Reference Manual for a detailed description of classes and functions.

1.1 Supported Platforms

The target for the NWC4 runtime libraries is the actual CTR hardware.

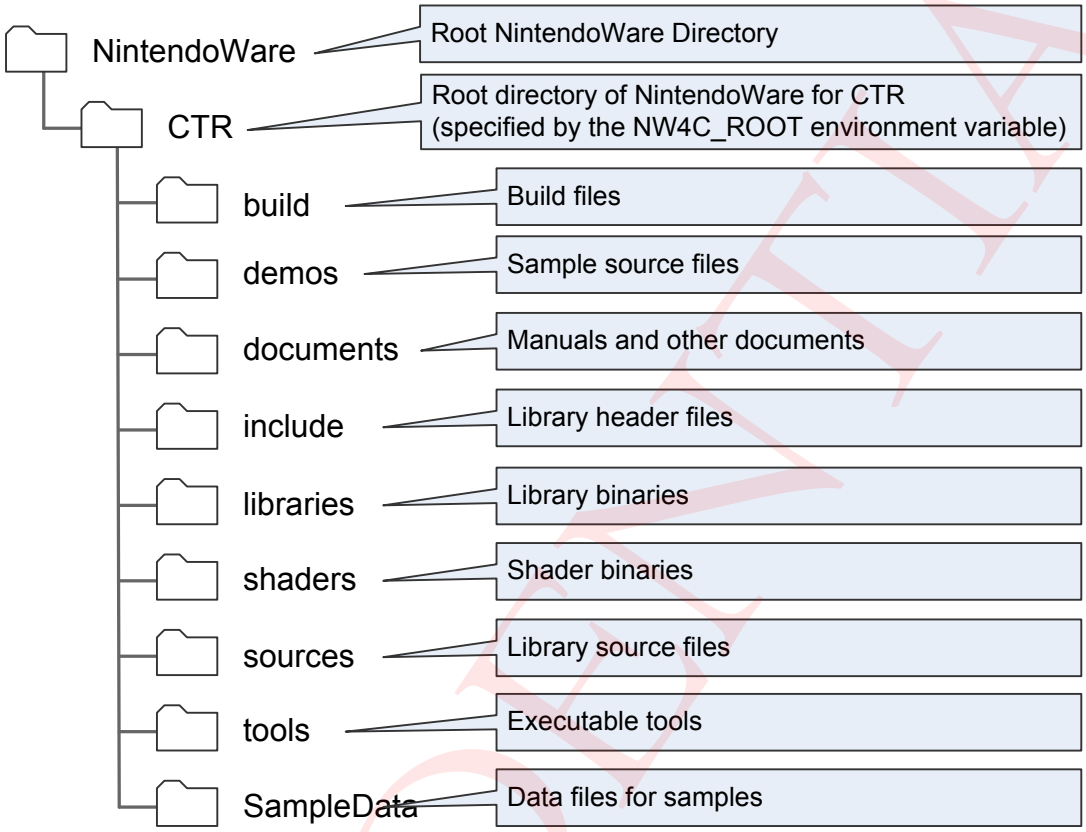
Note: The current version of the package does not include the PICA on Desktop (POD) graphics emulator, which runs on Windows and was provided as a PC-based development environment in preparation for the CTR target board development environment.

1.2 Environment Variable Settings

Figure 1-1 shows the NW4C directory structure.

Set the `NW4C_ROOT` environment variable to the absolute path of the `NintendoWare/CTR` directory. You can set the `NW4C_ROOT` environment variable automatically by running `NintendoWare/CTR/setup.wsf`. This directory is indicated by `$NW4C_ROOT` throughout the rest of this document.

Figure 1-1 Directory Structure



2 Quick Start

This chapter uses sample demos to explain how to run the NW4C runtime libraries on the CTR target board.

2.1 Preparing the CTR-TEG2 Development Environment

Follow the *CTR Startup Guide* to set up the development environment. You must be at the stage where you can build and run the CTR-SDK sample demos before you continue reading.

2.2 Building and Running a Sample Program

2.2.1 Building the Sample Program

Sample programs are provided in the `$NW4C_ROOT/demos` directory. First, run `OMake` in the `$NW4C_ROOT/demos/gfx/SimpleDemo` directory.

```
$ cd $NW4C_ROOT/demos/gfx/SimpleDemo
$ omake
```

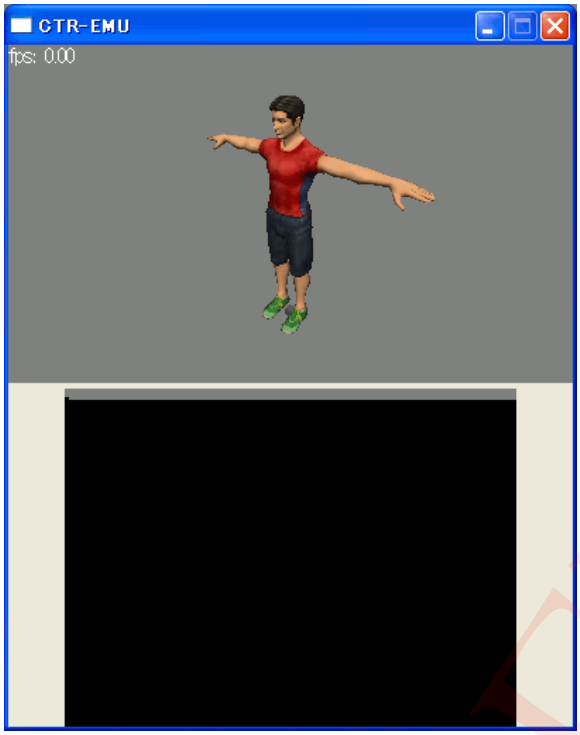
When the build succeeds, `SimpleDemo.cci` is created in the `SimpleDemo/images/CTR-TS.Process.MPCore.fast/Development` directory.

2.2.2 Running the Sample Program

Load the generated file, `SimpleDemo.cci`, in the PARTNER-CTR/S debugger. From the PARTNER-CTR/S debugger's **File (F)** menu, select **Load (L)** and open `SimpleDemo.cci`.

Next, from the **Run (R)** menu select **Run Program (G)** to run the sample program. If it runs properly, screens that resemble Figure 2-1 are shown on the LCD screens of the CTR target board..

Figure 2-1 Simple Demo Sample Screens



Note: This image was captured by running the program on a Windows PC computer and is different from the image that is actually displayed on the CTR target board.

3 Library Organization

This section describes the structure of the runtime libraries.

3.1 File Structure

The NW4C runtime libraries comprise the following types of files.

- Header files
- Library files
- Source files

3.1.1 Header Files

Header files are stored in the `$NW4C_ROOT/include` directory. To access the NW4C header files, the `$NW4C_ROOT/include` directory must be in your include path.

3.1.2 Library Files

Library files are stored in the `$NW4C_ROOT/libraries` directory. The `libraries` directory has the following subdirectories.

```
libraries/Target Platform/Build Type
```

You must specify the library path according to the target platform and build type that you are using.

3.1.3 Source Files

Source files are stored in the `$NW4C_ROOT/sources/libraries` directory.

Note: Behavior is not guaranteed for library files generated by a rebuild.

3.2 Kinds of Libraries

The NW4C runtime libraries can be broadly divided into the four following kinds of libraries.

- System libraries
- 3D Graphics libraries
- 2D Layout libraries
- Sound libraries

3.2.1 System Libraries

A number of libraries collectively called the System libraries are provided for use with the NW4C runtime libraries. These include the ut library, which defines by the utility class, and the io library, which defines the file input/output class.

3.2.2 3D Graphics Libraries

These are libraries for rendering to the actual hardware using the 3D graphics data exported from NW4C CreativeStudio, which is a Windows PC application.

These libraries support not only 3D models but also particles.

3.2.3 2D Layout Libraries

These are libraries for drawing to the actual hardware using the 2D layout data exported from NW4C LayoutEditor, which is a Windows PC application.

3.2.4 Sound LIbraries

These are libraries for playing audio on the actual hardware using the sound data exported from NW4C SoundMaker, which is a Windows application for sounds,

4 Revision History

Version	Revision Date	Description
1.7	2010/07/29	• Changed the document format.
1.6	2010/06/28	• Made revisions in step with end of support for TEG2 • Changed short name for NintendoWare for CTR to NW4C
1.5	2010/05/03	• Made revisions in step with end of support for PC runtime libraries
1.4	2010/04/15	• Made revisions in step with support for CTR-TS
1.3	2010/02/19	• Made revisions in step with addition of sample data folder
1.2	2010/01/15	• Revisions that accompany the sound library's CTR-TEG2 support. • Revisions that accompany renamed solution files for sample demos.
1.1	2009/11/10	• Added descriptions related to CTR-TEG2 support. • Revised descriptions related to environment variable settings.
1.0	2009/10/30	• Initial version.