

RPG Maker 20XX Engine

Written by WolfCoder

February 11, 2011

This software is licensed under the *GNU Lesser General Public License* (<http://www.gnu.org/copyleft/lesser.html>). Uses *Simple DirectMedia layer* (<http://www.libsdl.org/>). Uses DirectX 9. Uses *FMOD* by Firelight Technologies (<http://www.fmod.org/>). Compatible with *RPG Maker 2003* by Enterbrain, inc. All original graphics, text, and other materials copyright by WolfCoder (2011).

Permission is given to redistribute this document in an unmodified form from the original.



music & sound effects system



Links:

Dream.In.Code: <http://www.dreamincode.net/>

RPG Maker .Net: <http://rpgmaker.net/>

Meridian Dance: <http://meridiandance.org/>

Table of Contents

Introduction	4
Installation	6
Basic Features	8
Normal Mode VS. 20XX Mode	12
Special Commands	19

Introduction

The RPG Maker 20XX Engine is an open source replacement engine for Enterbrain's runtime engine for the RPG Maker 2003 Engine. It is designed to vastly improve upon the original system and to provide additional features. This engine can be used in other game projects and systems, not just RPGs and RPG Maker 2003.

Features include a more robust and stable sound engine, Direct3D rendering, full color, a user-based window drawing system, and more.

The original RPG Maker 2003 featured an active time-based battle system similar to that found in Final Fantasy 6. This provided more gameplay options. However, the implementation of the system is heavily flawed. In The RPG Maker 20XX Engine, multiple styles of battle systems can be used and switched in the middle of gameplay.

You must be familiar with RPG Maker 2003 or other tool before using this engine. This instruction manual will only cover the new features added with this engine.

What is SDL?

SDL is short for *Simple DirectMedia Layer*. It provides the basic window and video surface for many open-source applications dealing with graphics including emulators and video players.

What is FMOD?

FMOD is a widely used hobby and commercial sound engine for a large family of audio file formats. It provides an audio experience superior to that of RPG Maker 2003.

Installation

The files included are the engine files. There should be many .DLL files, many .EXE files and some extra resources. These files are to be copied into the folder of the target game for installation.

Installation for RPG Maker 2003

The engine files should be placed where you find RPG_RT.exe in the game folder. You may decide to keep RPG_RT.exe or delete it. You may also delete HARMONY.DLL, it is not needed for this engine.

Renaming the engine to RPG_RT.exe to replace the old one will allow you to start RPG Maker 20XX directly from the editor using the normal test play button.

Engine Versions (The Different Engine Files)

There are different .exe files to choose from. You may copy whichever version best fits your needs:

wcrpg2kxe.exe – This is the basic engine. Use this file if you're not sure which engine version you want.

wcrpg2kxe.wide.exe – This is the widescreen edition of the engine with an internal resolution of 400x240 instead of 320x240 pixels. You will need to add file replacements in order to support the widescreen format with files such as battle backgrounds and title cards.

Basic Features

There are a few basic features that are available at any time while using the RPG Maker 20XX Engine.

Controls

The following controls are the default for the RPG Maker 20XX Engine:

Arrow Keys:	Move Player
Z, Return, and Spacebar:	OK / Examine
X, C, V, B, and N:	Cancel / Open Menu
ALT+1, ALT+2, ... ALT+9:	Set Video Size (1x, 2x, ... 9x)
ALT+ENTER:	Toggle between Window/Fullscreen

The video size setting will be remembered for all games played in RPG Maker 20XX, so you only need to press the key combination once.

32-Bit Full Color Images

The RPG Maker 20XX Engine can use up to 32-bit color image resources with an optional alpha transparency channel in the following formats:


.CUR, .ICO, .BMP, .PNM, .XPM, .XCF, .PCX, .GIF, .JPG, .TIF,
.PNG, .TGA, .LBM, .XV, .XYZ


If you're using the engine with RPG Maker 2003, you must have a 256-color image compatible with the software and the 32-bit image in the same resource folder.


In order to use a 32-bit image, place the 32-bit copy of the image next to the original with -2x after the name but before the extension. If you


are using the RPG Maker 20XX Engine with a different program, you do not have to do this.

Example:

 Swamplands.png

 Tree.png

 Volcano.png

 Volcano-2x.png

Sample File Names

Here, *Volcano.png* is the file used when editing the RPG, but *Volcano-2x.png* is the actual 32-bit image to be used when the game is run.

Test Play

Test play in RPG Maker 20XX is handled differently than before. Test play is when the engine is started using the test play button in the editor. The following function keys are used during test play.

F5 - Opens the save menu so that the tester can save anywhere.

F6 - Takes a picture and saves it with the game save data.

F7 - Opens the game save folder in the file browser.

F9 - Sets the engine speed to normal.

F10 - Slows down the engine to test split-second timing.

F11 - Greatly speeds up the engine to race through scenes and events.

F12 - Toggles FPS and information overlay (useful for taking clean screenshots).

Normal Mode VS. 20XX Mode

The RPG Maker 20XX Engine can operate in 20XX mode, which is ideal for new projects using this engine. Add the following line to RPG_RT.ini or wcrpg.ini to enable this mode for your game:

```
Year=20XX
```

This will enable 20XX mode for your project.

Special Function Calls

The first six variables in the game will become the special function call variables. In order, they are labeled as A, B, C, D, E, and F. First, you write any parameters into A, B, C, and D (which means any special command will use up to 4 options). Then you write the number of the

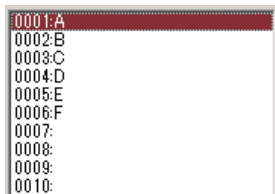
command you want to use into E. Then, you write the value 1 into F in order to execute the command.

You can re-define where these six variables will be by adding the following line in wcrpg.ini or RPT_RT.ini:

```
RegBase=#
```

Where # is the first variable of the six variables to be used for special function calls. The default for this would be 1.

In RPG Maker 2003, you can use the Variable Operations command to do this. None of the variables are cleared or modified except for F, so you can call a function again with the same options by writing 1 into F again.



A screenshot of a software window titled "Special Variables". The window has a dark red header bar. Below the header, a list of variables is displayed in a monospaced font. The first variable, "0001:A", is highlighted with a red background. The other variables are "0002:B", "0003:C", "0004:D", "0005:E", "0006:F", "0007:", "0008:", "0009:", and "0010:". The window has a standard grey border and a scroll bar on the right side.

0001:A
0002:B
0003:C
0004:D
0005:E
0006:F
0007:
0008:
0009:
0010:

These Are the Special Variables

Skip Title Screen Menu

If there is no text at all in the entry for New Game under vocabulary/strings, then the default title screen will not appear and the game will immediately start.

Large Party Size

Simply add new party members past the old limit of 4. You can make a party that contains up to 128 party members, but keep in mind the limitations of the particular battle system you decide to use.

Message Box Extra Commands

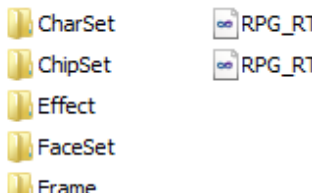
In RPG Maker 20XX, the `\v[#]` command can be nested many times inside other commands, and many special commands apply to all text inside the game such as item names, hero names, menu commands, strings, and more. `\c[\v[5]]` would color the text using a color referenced by variable 5, for example.

`\r[#]` will force a new line in any text where # is the number of lines to skip. Entering 0 will make the text reset to the left on the same line and a negative number places text lines above the current one.

HLSL Shaders

HLSL is the High Level Shader Language that is used with DirectX. Shaders tend to be tiny programs stored in text files ending in .fx. These programs are typically compiled and loaded inside of the video card while the engine is running. With shaders, many advanced techniques can be made. The system must support at least Shader Model 2.0 for any HLSL effect to work. Systems that support Shader Model 2.0 will also use an HLSL program for the screen tint effect which performs faster than the default software method.

In order to write and use an HLSL shader for your game, you must create a new resource folder named "Effect" and put all the shaders in there. The file name must begin with a number since this number is used with the actual shader commands in 20XX.



Example of Effect Folder Placement

Please refer to the documentation from Microsoft, tutorials, and shaders packaged with RPG Maker 20XX on learning HLSL. The following is only a list of global constants given to you by the engine.

texture wcdraw_fx_texture

This is the original target's image (tile, event, screen, etc.).
Sample from this texture if you want the sprite/picture fragment itself.

float4 wcdraw_fx_pixel_size

This contains the size of one pixel in sample space. The **x** and **y** components of this vector are the width and height of one pixel. If the original image was 32 pixels wide and 16 pixels tall, this vector would be 1/32 and 1/16 for **x** and **y**. Useful for doing pixel-exact effects (getting adjacent pixels)

float4 wcdraw_fx_tint_color

This contains the current screen tint color. The **r**, **g**, and **b** components of this vector are the red, green, and blue shifts. The **a** or fourth component of the vector is the saturation shift. All these values, on a normal screen, are 1.0 and range from 0.0 to 2.0.

Special Commands

Special commands are only available in 20XX mode, and they can greatly extend the abilities of the engine. There are three kinds of special commands: a command that add new features, commands that modify the database and commands that draw things onscreen.

Key:

(E) Name of Command: (A,B,C, and/or D for options)

Description of this command

System Commands

These commands extend the original features of event scripts so that they can do more. These are the simplest type of commands to use.

(0) Stop: ()

Immediately causes the engine to close, losing any unsaved progress in the game.

(16) Animate Enemies (a)

Monsters will use a battle character set instead if a battle character set has the same editor name as the monster. This animates the enemies so they act like heroes in the ATB battle system. This also effects anything that draws monsters.

(18) Re-Map Menu Command: (c,e,en)

Replaces a default menu command with a common event. C is the menu command to replace (see below for codes) and E is the ID of the common event to use. EN will enable the replacement if 1 and disable the replacement if 0. While enabled, the common event will be called every time the player selects the command from the built-in menu.

1 = Item, 2 = Skill, 3 = Equipment, 4 = Save, 5 = Status, 6 = Row, 7 = Order, 8 = ATB (Wait/Active Toggle)

(19) Set Self Variable Range: (s,e)

Self variables are bound to each event, so each event gets its own set of variables in this range. S is the starting ID and E is the ending ID of the variables to use for self variables. Changes will not take effect until a new map is loaded. Reading and writing to any of these variables will change the variable for the event and not the entire game as was the default. These variables are also not saved, they are always 0 when a new map is loaded by default.

(23) Open Sub-Menu: (m)

Opens a system submenu (like item or equip) directly. Also contains a method for opening the loading menu for loading a previously saved game in the middle of an existing game session. The following values for M are:

1 = Item, 2 = Skill, 3 = Equipment, 4 = Save, 5 = Status, 9 = Load

(26) Save Self Variables: ()

*Saves the self variables of the event that calls this function. This powerful command enables the event to hold its state even after saving and loading the game. **However, this increases the game save file and memory consumption for every instance of an event using this command.** To save space, ensure self variables you are not using for a particular event are set to 0. Use this command every time you want to save the event's current self variables, it will not automatically save them when they change. This command can make elaborate item events capable of being copy and pasted without having to assign new variables every time.*

(32) Load Data: (source,id,v)

Loads all data about the given object (referenced by id) into many variables starting at variable v. Be careful about variable ranges and

whether or not variables after v are used elsewhere. Source is the database section to load the data from (Monster, Item, etc.). Please check the 20xx-load-data-maps.txt reference for the complete list of how the data will be loaded. The following values for source are:

2 = Skill, 3 = Item, 6 = Monster

(33) Load List: (source,id,v,n)

Loads an element in a list from the database. Lists are typically things like skill lists, attribute effectiveness lists, and other variable length data in the database. ID is the id of the object in the database, v is the target variable to store the value in and n is the element in the database. If n is zero, the length of the list is returned, otherwise it returns the data in the list at that position.

0 = Hero (Current) Skills

Drawing Commands

These commands allow you to draw complex objects on the screen. They must be called inside of a Parallel Process or Auto Start event in RPG Maker 2003 so that these commands are called every frame. This is important as something needs to be drawn every frame for it to be visible. Likewise, you can stop drawing an object for it to disappear if this is what you mean to do.

(8) Draw Window: (x,y,w,h)

Draws a window using the current system graphic. X and Y are the location (in screen pixels) of the top left corner of the window. W and H are the width and height of the window in pixels.

(6) Draw Number (x,y,n,t)

Draws a number using the current system graphic. X and Y are the location (in screen pixels) of the top left corner of this number. N is the number to draw and T is the particular graphic to use. If T is 0, it

uses the number font out of the standard system graphic and if T is 1, it uses the battle gauge graphic.

(12) Draw Health Bar (x,y,p,w)

Draws a health bar that is W pixels wide in the actual bar area (not counting the ends), and is filled with P pixels. If P is W, the bar appears full, and if P is zero, it appears empty. X and Y is the location to draw the bar at. The bar is taken from the first bar out of the system gauge graphic.

(13) Draw Mana Bar (x,y,p,w)

Has the same behavior as Draw Health Bar, but instead draws the second bar in the gauge graphic.

(14) Draw ATB Bar (x,y,p,w)

Has the same behavior as Draw Health Bar, but instead draws the third bar in the gauge graphic.

(20) Draw Battle Character (x,y,id,p)

Draws a battle character, animation and all. X and Y is the location (from the top left of the cell, not the center), id is the ID of the character to draw from Animations 2 in the database, and p is the pose number. Refer to the database to find these two last numbers if you're confused.

(21) Draw Battle Attack Right (x,y,id,w)

Almost the same behavior as Draw Battle Character, except this draws the character attacking with weapon w using their right hand.

(22) Draw Battle Attack Left (x,y,id,w)

This is the same as Draw Battle Attack Right, but for the left hand.

(24) Draw Monster (x,y,id,p)

Draws a monster at X and Y where ID is the ID number of the monster to draw. If the monster uses a battle character set instead, that

character set will be drawn and P will be the pose of the character set to draw.

(25) Battle Character Options (f,m)

Sets the options for all future drawn battle characters. If F is non-zero, it becomes the frame number to display instead of the battle character animating itself. If M is non-zero, the battle character will face the other way and be flipped around.

(30) Draw Image (id,x,y,a)

Draws an image from a file name beginning with the number ID located in the "Picture" material folder. X and Y is the location to draw at (starting from the top left corner of image file), and A is the transparency level from 0 to 100 with 100 being completely invisible.

(31) Draw Name (source,id,x,y)

Draws the name of an object from the database at X and Y on the screen. ID is the ID number of the object in the database (item, hero, etc.). The following source codes are used:

0 = Hero Name, 1 = Hero Title, 2 = Skill Name, 3 = Skill Description,
4 = Item Name, 5 = Item Description, 6 = Monster Name, 7 =
Attribute Name, 9 = String From Vocabulary (See separate document
for the list of string codes)

HLSL Commands

These commands let you assign HLSL shaders to various things. These commands will not work if the target machine does not support Shader Model 2.0 (and some HLSL programs can be written for higher shader models), so prepare some basic fallbacks in the case that all shaders are disabled.

(27) Set Screen Shader (s,e)

Sets the shader to be applied to the whole screen after all the basic objects are drawn (objects below the menu layer). Useful for environment, weather, and lighting effects. S is the shader number to use (the file name will also begin with this number, see the section on HLSL under features) and E is the enable value. If E is zero, this will disable the screen shader and return the screen to normal.

(28) Set Event Shader (s,e)

Identical to the Set Screen Shader command, but effects only the event running this command (this event).

(29) Set Tile Shader (s,e,t)

Identical to the Set Screen Shader command, but effects specific tiles referenced by T. All shaders set to tiles will become reset once a new map is loaded (via Teleport command or otherwise). Use the following tile codes for T:

0,1,2 = The three ocean animated autotiles

3,4,5 = The three animated tiles



RPG Maker 20XX Engine

18-161 = The lower tiles

162-305 = The upper tiles

Special Thanks To

Saline / Magi
Feldschlacht IV
Link_2112
Ocean
kentona
Drakonais
Archeia Nessiah
KEF