LUNA
Engine

INTRODUCTION

It all started back when RPG Maker VX was released. My friends and I would always lament how it's impossible to just drag pictures and make our mockups work like magic. No need for script hunting or bug programmers to do every miniscule edit to our projects because we couldn't do it. With RPG Maker VX Ace and a small conversation in a private chatroom, we finally started work on it.

The Luna Engine is a project by Archeia_Nessiah and Yami, with the amazing help of NeonBlack and Rhyme. It is to create a series of scripts that will allow people to modify the GUIs present inside the game, whilst being easy to use and compatible with the majority of scripts. The Luna Engine took us a little more than a year (June 15 2013~July 2014) to develop, put through beta, make sample graphics for and write the manual. While it was a long journey, it was definitely worth it.

Due to its nature as a GUI script, it might conflict heavily with scripts that do numerous visual modifications. The Luna Engine works best for vanilla battle systems such as the Default Battle System or Yanfly's Battle Engine due to its popularity, added features and user friendliness. Because of Neonblack's involvement with this development, it should work with his CP Battle System. This should work with Sideview Graphical Enhancement Scripts like Battle Engine Symphony and Kread's Animated Battlers.

We do not have plans at the moment to make it compatible with any other scripts *aside from* Yanfly, Yami, Neonblack and Kread's scripts. Please don't ask us for compatibility patches for any other systems because we won't be able to provide them. We might attempt to add more compatibilities in the future, but for now, it is not possible.

If you would like to report bugs, please remember the following things before reporting:

- You are required to at least be familiar with RPG Maker's default function. An example scenario would be, if your test character's skills aren't displaying, then check their levels, etc. We had instances where people thought it was Luna Engine's fault.
- Make sure to test the scripts without other scripts to be sure it's the Luna Engine's fault.
- Make sure to read this manual and/or reference it. Use CTRL+F or Find for any questions or about features you're uncertain of.
- You can ask for tech support at rmlunaengine@gmail.com if this manual doesn't answer your questions or if you found a bug.
- For further updates in regards to the Luna Engine, you can check our Github or check the Luna Engine tag in Division Heaven.

INSTALLATION

1. Open the Script Editor (F11). Remember that CTRL+F only searches for items inside the present script. CTRL+SHIFT+F means it will search all scripts.

2. The script order is very important in RPG Maker. All Animated Battler scripts should be *below* the Luna engine. Simply install the scripts in this order. The following script order is just the bare necessities you'll need.

    - **Window Identifier -** This serves as a debug script and should only be used when modifying your scenes. Refer to the Basics for more information. To quickly enable/disable scripts, just CTRL+A (highlight all) and then press CTRL+Q.
    - **Configuration and Requirement Scripts -** The order doesn't matter but it would be wise to arrange them based on how they appear in-game.
    - **Lunatic Configuration Scripts -** Optional if you have no plans to use Lunatic.
    - **Battle Luna Core**
    - **Menu Luna Core**
    - **Battle Luna Add-ons**
        - Actor Commands
        - Battle Popups
        - Image Popups
        - Escape Command
        - Skill & Item List
        - Disable Party Commands
        - Circular Bars
        - Shaking HUD
        - Lunatic Face
    - **Menu Luna Add-ons**
        - Grid Status
        - Lunatic Backgrounds

3. If you want to add more features, refer to our main project folder and copy and paste scripts from there. Make sure to take note of the order of the scripts!

4. I recommend using Notepad++ (Free even plugins) or Sublime Text (Shareware) when editing scripts. They have flexible features and it could help you fix anything wrong with your configuration while checking for other errors during testing.

5. If you want to compare configurations with the demo scripts, Notepad++ has a plugin for it. Sublime Text has Sublimerge but costs $20. DiffNow is a personal favorite.

THE BASICS

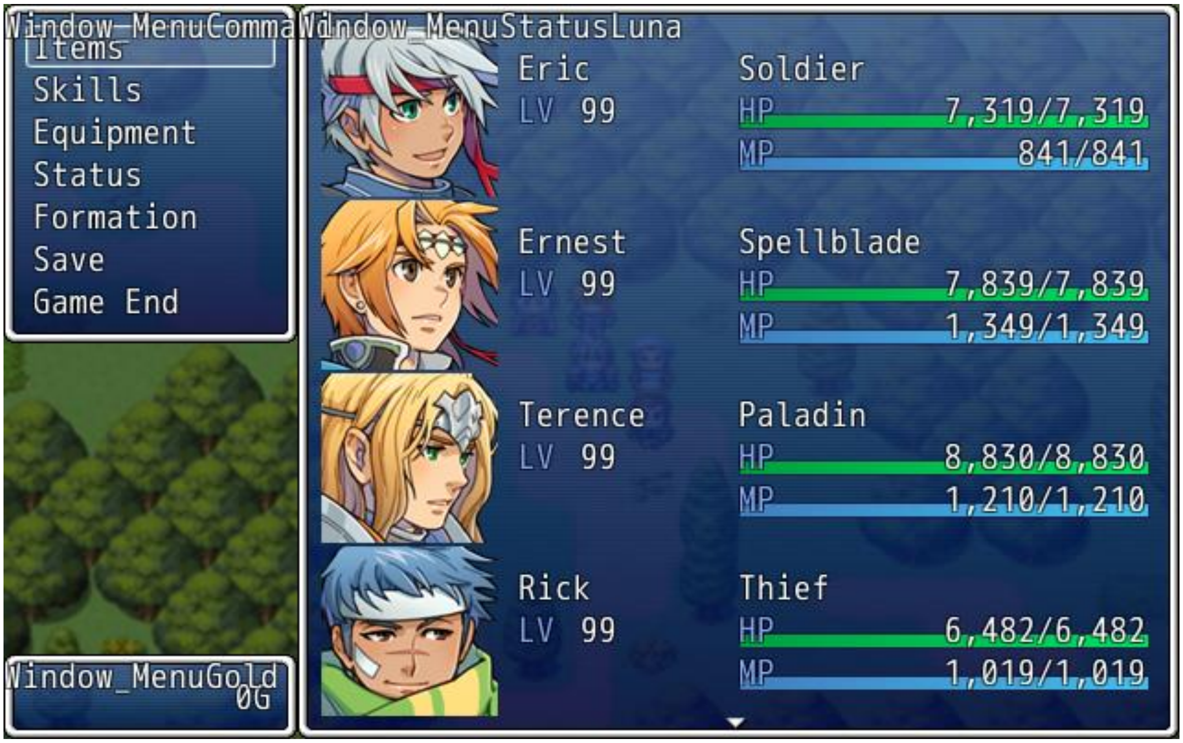
It is extremely important not to skip this section as this explains how the backend of RPG Maker works.

THE BASE

In order to use the Luna Engine efficiently, it’s important that you should know about *Scenes* and *Windows*. Imagine the Scenes as rooms of a huge building. The contents of the room varies depending on its purpose. A window, on the other hand, reveals the contents of the room. How much of the room is revealed depends on its size. This is how RPG Maker displays the items that the player needs to see.

So how is this related to RPG Maker? Basically, each menu has specific scenes (e.g. Scene_Equip, Scene_Status) and they all have their own windows while some have similar names for consistency. As long as you keep this in mind, you will be able to modify your game’s GUIs.

Rhyme has kindly provided us with a script that shows the name of the windows in a scene. It is named as Window_Identifier inside the game project. Just hold CTRL inside the game and you’ll be able to see the name of the Windows like this:



Inside the Luna Menu scripts, just remember to search for the nearest name (e.g Status_Window could represent Window_MenuStatusLuna).

RESOURCES

All your GUI resources should be located in the Graphics/System of your project folder. Everything else stays the same (e.g. Facesets still go to Facesets). You can name them however you want, there should be no issues.

GENERAL OPTIONS

The list below just refers to the general options of the configuration script. It doesn't cover everything and serves as a refresher or a short course of the options. Most, if not all, options found in the script are commented, so there should be no problems figuring out what they can do in case the manual doesn't cover it.

- **:width** – the width of the window or item.
- **:height** – the height of the window or item.
- **:max_width and :max_height** – these help the system find a value to center the contents shown in the screen.
- **:x** – this refers to the X coordinate of the screen. This refers to the horizontal (left and right) axis. Negative (e.g. -1) values mean they would move to the left, while Positive values mean they would move to the right.
- **:y** – this refers to the Y coordinate of the screen. This refers to the vertical (up and down) axis. Negative (e.g. -1) values mean they would move to the up, while Positive values mean they would move to the down.
- **:offset_x and :offset_y** – these refer to how much you want to nudge the window display without affecting its base x and y. Useful for items such as GUI borders.
- **:z** – this refers to the item's display priority. Think of it as layers when you use an image program or when mapping in RPG Maker. The higher the value, the higher it will be drawn above other items.
- **:offset_z** – this makes the item adjust itself by a specific value just in case you have drawn other items and want an item to remain on top.
- **:padding** – this refers to the spacing between the window border and the window contents. The default value is 12.
- **:spacing** – this function is the distance between the objects. Can be negative or positive values.
- **:vertical** – this helps the system determine if the cursor and contents should be displayed vertically if true, or horizontally if set to false.
- **:cursor** – this is to enable or disable the default cursor.
- **:lunatic** – when set to true, it will check the Lunatic configuration the user has set for that specific scene.
- **:background_variable** – this option allows you to have multiple skins for your GUI. Remember that it reads it as a string (e.g. $game_variables[1] = "Menu_Green") and set a different variable for each menu.

Another important thing to remember is that enemies and actors are referred to as Battlers in RPG Maker. So if you want to change anything related to the actor itself, it's most likely in Battler_Status section of Luna's Configuration Script.

THINGS TO REMEMBER

- If you are confused about what scripts to copy for the Luna Engine, either make a copy of Luna Engine Basic Base and modify it or copy the scripts from there and transfer to your project.
- Variable 1 is used by the Luna Engine scripts. Change :background_variable's value to something else if you're using it already.
- If you want to use Sideview Battle Systems, you should set animation_on_hud configuration to false.
- When using Yanfly's Free Turn Battle (FTB) set :arrow_battler in YEA Compatibility to true and if use spin command, :all_dir to false.
- If you want to use the horizontal actor command, set :arrow_battler in YEA Compatibility to false.
- If you want to change the battler options and it doesn't work, try setting lunatic to false. If you are using lunatic, read the comments as you have to draw the names manually.
- If your controls aren't working properly (e.g. Horizontal menu inputting up and down instead of left to right) try adjusting **:col_max** value. Remember that it has to be in synch with **:limit_page.**
- If you're using lunatic for the character display in Main Menu Luna, make sure to enable :collapse for :main as true to enable grayscaling a dead actor's portrait.
- If you want to have more than four party members, go to *Adjust Party Size by Archeia.*
- If the game auto closes, don't forget to add TRGSSX.dll in your game's folder.