LEMoon Setup for Visual Studio 2017 Community

1. Download **Visual Studio 2017 Community** from this website and install it: https://www.visualstudio.com/de/downloads/

2. After installing VS 2017 download the **Lynar Moon Engine** from **gitHub**: https://github.com/LynarStudios/LEMoon → press the **Releases** tab and get the latest release as **.zip** file or **tar.gz** file.

2.1. Since this whole tutorial is in English it is highly recommended to first change the language of **Visual Studio 2017 Community** to English! You can do that by clicking **Tools → Options → Environment → International Settings → Language → English → OK**
To apply these changes you need to close and open **Visual Studio 2017 Community** completely again.

3. Open **Visual Studio 2017 Community**! Then click **File → New → Project**. Choose **Installed → Visual C++ → Empty Project**. Choose an appropriate name for your project and enter it in the **Name:** field (acronym: **PN** for project name). And remember your **Location:** folder (acronym: **LF** for location folder)! This is very important! Make sure that the checkbox for **Create directory for solution** is checked! Press **OK** then.

3.1 In case you want to contribute to any official **Lynar Studios Project** you need to set some advanced settings to stick to the **Lynar Studios** code style. Otherwise you can skip this step.

Click **Tools → Options → Text Editor → C/C++ → Tabs** and make sure that the following settings are set:

Indenting → None
Tab → Tab size → 1
Tab → Indent size → 2
Tab → Insert spaces

Click **Tools → Options → Text Editor → C/C++ → Formatting → General → When I paste → Do nothing**!

Click **Tools → Options → Text Editor → C/C++ → View → Outlining** and make sure that the following settings are set:

Enable Outlining → True
Outline Pragma Regions → False
Outline Statement Blocks → False

Press **OK**!

4. Now unpack the whole **Lynar Moon Engine** (LEMoon-x.x) folder to this location: **LF/PN/PN**. This directory should exist already!

   e.g. **C:\Users\USER\source\repos\LEMoonTut\LEMoonTut** on my machine where **USER** is the windows user name, **C:\Users\USER\source\repos** is **LF** and **LEMoonTut** is **PN**.

   If this directory doesn't exist already you did something wrong. If you did everything correctly there will be a **LEMoon-x.x** folder in this directory after you unpacked it! e.g **C:\Users\USER\source\repos\LEMoonTut\LEMoonTut\LEMoon-x.x**

5. Now right-click **Solution Explorer** → **PN** → **Header Files** → **Add** → **New Filter** in **Visual Studio 2017 Community** and call it **LEMoon**. Right-click on this new filter then and choose **Add** → **Existing Item...** Select all header files (not the **glm** folder), which are located in **LEMoon-x.x/include** and click **Add**.

   Now right-click **Solution Explorer** → **PN** → **Source Files** → **Add** → **New Filter** and call it **LEMoon** as well. Right-click on this new filter then and choose **Add** → **Existing Item...** Select all source files, which are located in **LEMoon-x.x/src** and click **Add**.

6. Now make sure that in **Solution Explorer** **PN** is selected! Select **Project** → **Properties**. A Pop-up window should appear! Now set **Configuration:** to **All Configurations** and **Platform:** to **x64**, since you want to build **64-bit** applications nower days. Of course it's possible to build **x86** applications with **LEMoon** as well. Leave this window open now.

7. Select **Configuration Properties** → **VC++ Directories** and leave this option open! Now select **Include Directories** → **<Edit...>**. A new Pop-up window should appear. Press the button for **New Line** and press the **...** button. Now navigate to your **LEMoon-x.x** folder and select the **lib/SDL2/include** directory and choose it. Do this with **SDL2_image**, **SDL_mixer** and **SDL_ttf** as well. All these directories contain **include** directories. You should have 4 entries in the list then. Press **OK** then.

   Select **Library Directories** → **<Edit...>**. Press the button for **New Line** again and press the **...** button. In your **LEMoon-x.x** folder select **lib/SDL2/lib/x64** and choose this directory. Do the same with **SDL2_image**, **SDL_mixer** and **SDL_ttf** again. All these directories contain **lib/x64** directories. You should have 4 entries in the list then. Press **OK** then and **Apply** these changes.

8. Select **Configuration Properties** → **Linker** → **Input** → **Additional Dependencies** → **<Edit...>**. A Pop-up window should appear. Now enter these 5 lines in the input field:

   SDL2.lib
   SDL2main.lib
   SDL2_image.lib
   SDL2_mixer.lib
   SDL2_ttf.lib

   Press **OK** then and **Apply** these changes.

9. Select **Configuration Properties → Linker → System → Subsystem → Console (/SUBSYSTEM:CONSOLE)** and **Apply** these changes.

10. Select **Configuration Properties → C/C++ → Preprocessor → Preprocessor Definitions** and enter these to lines:

    _CRT_SECURE_NO_DEPRECATE
    _CRT_NONSTDC_NO_DEPRECATE.

    Press **OK** and apply these changes then.
    You can now close this Pop-up window by clicking **OK** again.

11. The **Lynar Moon Engine** is now fully set up. Add a **main.cpp** file now to your project by right-clicking **Solution Explorer → Source Files → Add → New Item...**
    This file should contain a **main function** definition with this prototype:

    int main(int, char);**

12. Make sure that at the top of the main window of **Visual Studio 2017 Community** **Release** and **x64** are selected too. Otherwise you will get some compile errors!
    Also make sure that in **le_glb.h** **LE_WINDOWS** is set and **LE_LINUX** and **LE_ANDROID** are commented out (//);
    Click **Build → Build Solution** then. This will compile the **Lynar Moon Engine** for you and is then ready for use.

12.1 At last you need to include the appropriate **.dll** files. You will find these files in the **x64** directory of **SDL2**, **SDL2_image**, **SDL2_mixer** and **SDL2_ttf**.
    e.g. **LEMoon-x.x/lib/SDL2/lib/x64**

    All **.dll** files need to be included to the **LF/PN/x64/Release** directory. This is the same directory where you should find the **.exe** file that has been created after compilation.