A Unity tool by Jesse Stiller

I need to contact the creator of this tool

Sure thing. Here's my email address: aquageneral@hotmail.com.

What is Phlayer?

Phlayer makes working with layers attached to game objects fast, safe, and automatic (in more ways than one).

Layers in Unity are very powerful; they can be used to differentiate objects about as quickly as possible for a computer. And they are widely used in physics engine queries such as raycasts.

There's a very big problem with layers in Unity without Phlayer, however: they are stored internally in an asset which can only be accessed indirectly. Not only is this slower than how it can be, but there can be outdated references which can cause painful bugs to arise, and more.

Phlayer solves this by automatically generating a script containing all of the layers each time they are edited in the Layer Inspector. Here are all the benefits that Phlayer brings:
Robustness - compiler time errors will be generated when layers have been changed but code has not. There won't be any unexpected bugs to track down

How do I setup Phlayer?

Simply import Phlayer into your project and you're already finished setting up Phlayer. Make sure you do not rename Phlayer's files or directories since Player relies on fixed names.

How do I use Phlayer?

Phlayer automatically generates a C# class that contains all of the layers in the current project every time the Layers are edited (in Edit > Project Settings > Tags & Layers).

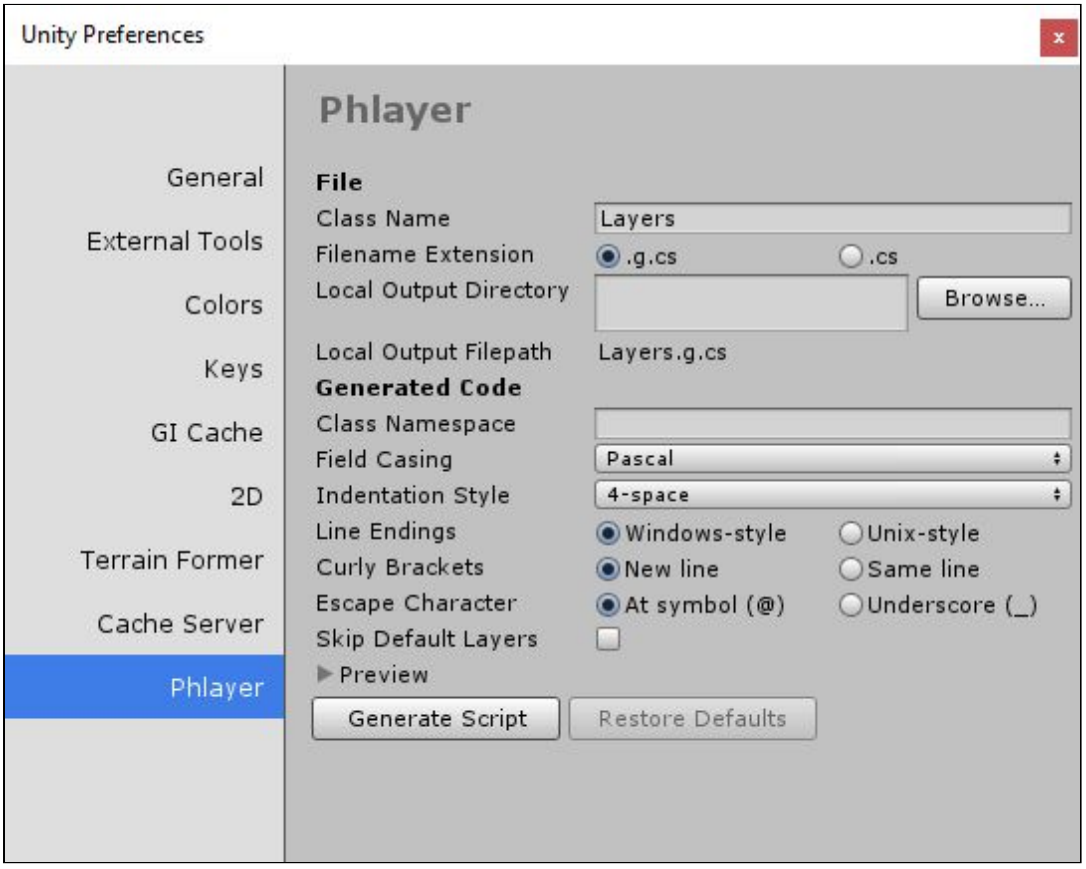
As long as Phlayer is imported into your project and you haven't modified any file/directory names, Phlayer will always run automatically.

You can edit the coding style of the layer's class that Phlayer automatically generates by clicking Edit > Preferences and clicking the "Phlayer" section. While you are here, you can preview your changes by expanding the "Preview" dropdown, and you can also forcibly tell Phlayer to update the generated file by clicking "Generate Script".

What do the settings mean?

Note that all settings don't have a "best setup", but whatever your or your project's coding style determines the best settings to choose. If you are working on your own, it's okay to leave these settings to their default values—everything will work fine regardless.

All "Generated Code" settings are set by default to Microsoft's coding style used on MSDN.



File	
Class Name	The name of the generated layers class. This affects the C# name and the filename.
Filename Extension	The filename extension for the generated layers class.
Local Output Directory	The directory that the generated layers class will be saved to. This is relative to [Your Unity Project Directory]\Assets\
Local Output Filepath	This is a preview of the exact local filename of the generated layers class.

Generated Code	
Class Namespace	The C# namespace that the generated layers class uses. Namespaces essentially act like folders for layers.
Field Casing	The casing type used for the layers in the generated layer class. You can preview the differences this makes by expanding the Preview foldout to see exactly what does what.
Indentation Style	The indentation style to use for the generated layers class. The default value is a safe choice; the only right value is determined by whatever your or your project's coding style is.
Line Endings	This changes the way line endings are encoded in the generated layers class. Windows internally uses the character sequence "/r/n" to say "there is a new line here", while Unix style is "/n". This only makes a true difference if your file versioning system (if applicable) expects certain line endings.
Curly Brackets	Should the curly brackets be placed on a new line?
Escape Character	If there is a layer name that uses a reserved or otherwise invalid C# identifier, then this escape character will be used. An example of this could be the "Default" layer. The word "default" is a keyword in Unity and therefore this will be called "@default" if camel casing is used.
Skip Default Layers	Should the generated layers class include the default layers that Unity internally uses, such as "Default" and "Ignore Raycast"?

Need any help? Email me at aquageneral@hotmail.com.

Written by Jesse Stiller.
Last updated 23rd of May, 2018.