Requirements

- C# Compiler (Recommended: Visual Studio Community 2017 or Newer)
- .NET Decompiler (Recommended: dnSpy)
- **uMod Framework**
- UMF API Documentation
- Unity Script Reference
- Harmony Documentation
- A game with UMF installed into.
- Some experience and/or patience and willingness to learn.

Mod Project Files

1. Run `UMF.ProjectGenerator.exe` in `\uModFramework\Tools\` and fill in the data, then click `Generate Project Files`.
    - UMF.ProjectGenerator_Example.png
    1. Mod name with no spaces or special characters. This is used for the project files, class and assembly name.
    2. Where to save the source code of the mod.
    3. If a game has more than one data/managed folder choose the appropriate one. (32bit/64bit/etc)
    4. Harmony is used to overwrite existing code or inject new code into existing functions.
    5. Unity Scripting is used to add new code to the game, the same as the game already uses. See the Unity Script Reference to learn about how MonoBehaviour GameObjects work.
    6. All References adds all dll files in the Managed folder as References, giving you access to all Unity classes as well as any separate classes the game provides. Only use if you know what you are doing.
2. Open your project solution by double clicking ModName.sln in the newly opened folder.
    - File:UMF.ProjectGenerator_SolutionFolder.png
    - File:UMF.ProjectGenerator_VisualStudio.png
3. Add your code and click Build > Rebuild. (Rebuild will clean your project before each build.)
    - See **Important Links and Info** to learn more about coding.

Important Links and Info

- **UMF API** - The UMF API gives you access to various functions that makes modding easier, and is also required in order for the mod to be started.
- **Mod Installer** - The UMF Mod Installer (URI Handler) lets you create one click Install links for your mods.
- **Harmony** - Harmony is a useful library provided with UMF that lets you overwrite and inject code in MEMORY into existing classes and functions.
- **UMF Patch** - An IL-based UMF scripting patch system used as a last resort to let you automatically overwrite code in dll files on disk when Harmony can't access it in any way.
- **Source Mods** - UMF can compile and run mods at runtime directly from .cs files placed in `\Mods\Source\ModName\`. (Requires Mono to be installed.)
- **Example Mods** - A list of open source UMF mods you can use as examples to better learn coding with UMF.
- **dnSpy Guide** - A basic guide on how to use dnSpy for modding.

UMF starts and access mods through **Attributes**.
Mods can currently be provided as .cs files, .dll files, .zip files(containing .dll files), and .umfmod files(packed and encrypted) by the **Mod Packer**.
UMF and Mods made with UMF does not re-distribute any code or game files from games or the Unity Engine. Making them both morally and legally safe.
Mods made with UMF can also be freely sold/marketed by the Mod Creator. (Provided they don't break the UMF License)
