LESSONS LEARNED: LICENSING UNITY GAMES

EXACORE42

Abstract. After finishing development of Catanks, UPGRADE was interested in releasing the game under an open source license. However, due to lack of planning, this ended up being significantly harder than expected. This paper will document what exaCORE42 learned while licensing Catanks in the hopes that it will be useful for future UPGRADE games and other game development clubs.

⚠ **Warning**

I am not a lawyer. This is not legal advice, just my current best understanding of copyright law and licensing.

1. Choosing a License

There are many different open source licenses to choose from. When licensing Catanks, here is an overview of the options we considered.

License	Used?	Key Features	Reason Used/Not Used
GPLv3	❌	The GPLv3 license requires the entire project and any components to also be licensed under GPLv3 (or a compatible license).	Unity is not open source and thus, licensing any part of a Unity game under GPLv3 doesn't make sense because the terms of the license cannot be fulfilled.
MPL 2.0	✅	The MPL 2.0 license ensures that any modifications to MPL licensed files must also be licensed under the MPL 2.0 license, but puts no licensing requirements on other files.	We chose to license all UPGRADE-contributed script, shader, material, prefab, and animation files under this license because we wanted any modifications to our code to contribute back to the open source ecosystem. Additionally, the MPL license allows our open source code to interface with the closed source game engine (Unity)
CC BY-SA 4.0	✅	The CC BY-SA 4.0 license allows others to use the art but requires them to also license any modifications under the same license (or a compatible license). Additionally, it requires that anyone who uses the art to give credit for producing the	We chose to license UPGRADE-contributed art is licesed under the CC BY-SA 4.0 license because it allows our art to be used freely, but ensures that credit is given to UPGRADE's artists. Additionally, it ensures that any modifications to our art will

License	Used?	Key Features	Reason Used/Not Used
		original work to the authors of the orginal work.	contribute back to the open source ecosystem.

2. Before Development

2.1. License Agreement.

Before each person contributes code to the project for the first time, they should sign or agree to a licensing agreement that allows their work to be licensed under the chosen license(s). An example license agreement can be found at **Section 10**.

2.2. Project Layout.

Separating files by their type makes licensing much easier. For Catanks, an example of a folder layout could be:

1. `Assets/Images/original/`
2. `Assets/Images/reused/`
3. `Assets/Audio/original/`
4. `Assets/Audio/reused/`
5. `Assets/Models/original/`
6. `Assets/Models/reused/`
7. `Assets/Materials/original/`
8. `Assets/Materials/reused/`
9. `Assets/Shaders/original/`
10. `Assets/Shaders/reused/`
11. `Assets/Scripts/`
12. `Assets/Prefabs/`

3. Best Practices

This section covers the best practices to follow as a Unity game is being developed to ensure that licensing is not a difficult process. This section is mainly intended to help developers ensure that the content that they use is easily able to be licensed at the end of development.

Notification

If you want to use content from a previous UPGRADE game, ensure that the game has been properly licensed. So far, the only game that meets that criteria is Catanks. Further instructions about how to use previous UPGRADE content are in the sections below.

Warning

Even if a site advertises “royalty free” art or free to use code, oftentimes, that does not mean that one can release it under an open source license.

3.1. Code.

Do not use code that you do not have the rights to license under your chosen open source license, such as:

1. Code that was distributed as part of a course

2. Code that you found on a forum or tutorial
3. Code that was authored in part or in full by a person who has not signed the license agreement

See **Section 6** for cases when you would be allowed to use code that isn't your own and how to license it.

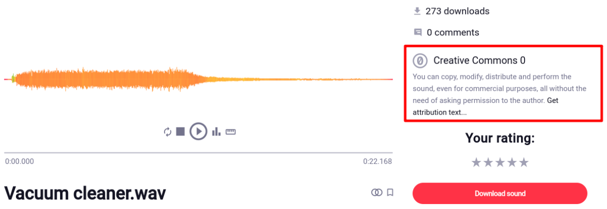
3.2. Art.

See **Section 7** for information about how to properly license art.

To use art (or shaders) from a properly licensed UPGRADE game, place the art in the directory `Assets/<type of art>/reused/<filename>`) and include a text file with a link to the file on Github in that directory.

3.2.1. *Sound Effects.*

1. If you are making your own sound effects from scratch place them in `Assets/Audio/original/`
2. If you are not making your own sound effects, only use sounds from freesound.org. Make sure that any sounds you use are licensed under (see **Figure 1** for where to look for this information):
   a) `Creative Commons 0`
   b) `Attribution 4.0` (CC BY 4.0)
   c) `Attribution 3.0` (CC BY 3.0) if needed, but preferably one of the above
3. If you use sounds from freesound.org as explained above, place each sound in its own folder (such as `Assets/Audio/reused/<name of sound>/`. In the folder, also include a text file that contains a link to the sound on freesound.org and a note about whether the original sound was modified.



273 downloads

0 comments

Creative Commons 0

You can copy, modify, distribute and perform the sound, even for commercial purposes, all without the need of asking permission to the author. Get attribution text...

Your rating:

0:00.000

0:22.168

Vacuum cleaner.wav

FIGURE 1. Licensing information for a sound on freesound.org

3.2.2. *Music.*

For Catanks, all the music was created by UPGRADE members and placed in the `Assets/Audio/original/` directory.

3.2.3. *Other Art.*

For other art including 3D models and 2D images, only use UPGRADE made art.

4. Licensing Tools: REUSE

In order to license Catanks, we used an open source tool called REUSE. REUSE made it easy to license all the files in the project and provides a tool to check compliance with the REUSE specification. This section will cover how to use the REUSE tool. This will be important to understand in the following sections.

REUSE is a system that makes licensing much easier by either including licensing information with the source files or specifying it in the `REUSE.toml` file. I highly recommend reading REUSE's tutorial and FAQ.

4.1. How does REUSE work.

The central idea of REUSE is to store all the licenses for the project in a single location: the `Licenses` directory. There are 3 ways for files to be licensed according to the REUSE specification:

1. Directly in the file using comment headers[1]

[1] Recommended method for source code files: https://reuse.software/faq/#why-care

2. Next to the file using `.license` files[2]
3. In the `REUSE.toml` file

4.2. Licensing Methods Used.

Licensing Method	Used?	Files Used For	Reason Used/Not Used
Comment Headers	✅	Most script files	Convenient for future users of the code and portable[1]
`.license` Files	❌	—	For every file in the `Assets/` directory, Unity adds a corresponding `.meta` file. If we added a `.license` file for nearly every asset file, Unity would add thousands of `.meta` files which would clutter up the repository. Additionally, Unity allows assets to be moved within it's editor easily, but doing so would leave behind old `.license` files.
`REUSE.toml`	✅	Art, shaders, some script files, material files, animation files, models, images, fonts	Allowed for bulk licensing[3] of files. Is a centralized location for tracking the many different licenses used in the project.

4.3. Comment Headers.

The general format for comment headers in `.cs` files is:

```
// SPDX-FileCopyrightText: <year> <copyright holder>
//
// SPDX-License-Identifier: <SPDX license>
```

However, one generally shouldn't add these comment headers manually. The REUSE tool has an easy way to add comment headers to files.

> **Notification**
>
> Currently, REUSE tool does not have a way to modify or delete headers, so be careful!

4.4. `REUSE.toml` File.

The format for a `REUSE.toml` file entry is:

```
[[annotations]]
path = [<list of files>]
precedence = "override"
SPDX-FileCopyrightText = [<list of copyright text>]
SPDX-License-Identifier = <SPDX license code>
```

The precedence doesn't have to be `override`, there are other options, but to keep things simple when licensing, you should use `override`. Check out Catanks' `REUSE.toml` for an example.

[2] https://reuse.software/faq/#uncommentable-file

[3] https://reuse.software/faq/#bulk-licenses

> ***i*** **Info**
>
> The last `[[annotation]]` in the `REUSE.toml` to cover a file is the one to define the licensing information for that file.[4]

For more information about `REUSE.toml`, see the REUSE specification.

5. Licensing Tips

5.1. Contributors as a Group.

6. Reusing Code

6.1. Using Comment Headers.

> ***i*** **Info**
>
> When licensing a file that uses someone else's code that you are allowed to use (see **Section 6**), make sure to include a `SPDX-FileCopyrightText` tag for the original author and a `SPDX-License-Identifier` tag for the license the original code was distributed under in addition to those tags for your contributions.[5] For example:
>
> ```
> // SPDX-FileCopyrightText: 2019 Sebastian Lague
> // SPDX-FileCopyrightText: 2024 The Catanks Contributors
> //
> // SPDX-License-Identifier: LicenseRef-MIT-PathCreator
> // SPDX-License-Identifier: MPL-2.0
> ```

7. Licensing Art

8. Reusing Fonts

9. Licensing Gotchas

9.1. TextMeshPro.

9.2. Configuration Files.

10. Appendix A (FINISH FORMATTING)

UPGRADE, Philadelphia, PA, 19104

[4] https://reuse.software/spec-3.3/#reusetoml

[5] https://reuse.software/faq/#copy-work