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 **Appendix A**.

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.

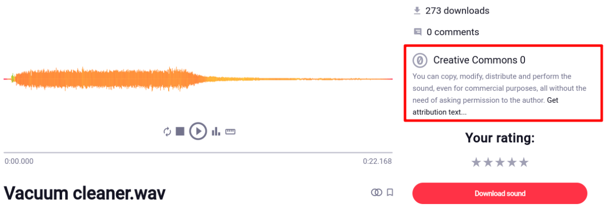
3.2. **Art.**

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 and a note explaining whether the file was modified (and if so, how).

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 (and if so, how).



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...

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.

3.2.4. *Fonts.*

Only use open source fonts. I recomment only using Google Fonts to find fonts (they only have open source fonts and it is easy to locate the fonts' license during the licensing process).

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, FAQ, and documentation.

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]
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.

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

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

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

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.

Tip

The path string(s) can use globbing to match many files at once.[4]

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.

4.5. `.gitignore`.

The REUSE tool ignores files that are not in the git repository and and in the `.gitignore` file.[5]

5. Licensing Tips

5.1. Contributors as a Group.

In Catanks, since we had many contributors, instead of having a copyright notice for each person who contributed to a file, we included a single copyright notice with the author as `The Catanks Contributors` and a file in the at the root of the repository called `CONTRIBUTORS.md` that listed all the contributors. This is a valid way to do copyright notices according to REUSE.[6]

5.2. Custom Licenses.

To use custom licenses with REUSE, one can place their own license in the `LICENSES/` directory with the name `LicenseRef-<name>.txt`. This license can be referenced with the SPDX license code `LicenseRef-<name>`.[7]

5.3. Configuration Files.

Copyright protection only applies to "orginal works of authorship"[8] where free choices were made about how to create the work.[9] Therefore, it follows that many config files present in Unity projects are not copyrightable.[9, 10] There are a few ways to deal with these config files. Catanks chose to license them under the license used for source code files, as explained by the advice given by REUSE

5.4. Comments.

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

[5] https://reuse.software/faq/#exclude-file

[6] https://reuse.software/faq/#many-copyright-statements

[7] https://reuse.software/faq/#custom-license

[8] https://reuse.software/faq/#what-is-copyrightable

[9] https://fsfe.org/news/2025/news-20250515-01.html

[10] https://reuse.software/faq/#uncopyrightable

If you want to include more information regarding how a license forr a certain file was determined, you can include this information in the `SPDX-LicenceComments` tag.

Example

```
[[annotations]]
path = [
  "Tanks/Assets/TextMesh Pro/Fonts/OrelegaOne-Regular.ttf",
  "Tanks/Assets/TextMesh Pro/Resources/Fonts & Materials/OrelegaOne-Regular SDF.asset",
  "Tanks/Assets/TextMesh Pro/Resources/Fonts & Materials/TitleText.asset"
  ]
precedence = "override"
SPDX-FileCopyrightText = "The Orelega One Project Authors"
SPDX-License-Identifier = "OFL-1.1-no-RFN"
SPDX-LicenceComments = "License found at https://fonts.google.com/specimen/Orelega+One/
license.  The link to the Orelega One Github page seems to be broken."
```

6. Reusing Code

6.1. What Code Are You Allowed To Use.

If you want to use someone else's code, you must ensure that the license they published their code under is compatible with the license your code will be published under. Additionally, certain licenses may come with extra stipulations

6.1.1. *MIT/BSD Licenses.*

Certain licenses such as the MIT and BSD licenses require those who use the code to keep a verbatim copy of the license text in their repository somewhere. Oftentimes, this text includes a copyright statement with the author's name. Catanks followed one of the options outlined by REUSE[11] and included the license text as a custom license and referenced it when necessary.

6.2. Licensing Code.

Notification

Make sure to identify **all** copyright holders in the copright text(s) when licensing (see sections below for examples).

6.2.1. *Using Comment Headers.*

When licensing a file that uses someone else's code that you are allowed to use (see **Section 6.1**), 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.[12]

[11] https://reuse.software/faq/#license-templates

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

Example

```
// SPDX-FileCopyrightText: 2019 Sebastian Lague
// SPDX-FileCopyrightText: 2024 The Catanks Contributors
//
// SPDX-License-Identifier: LicenseRef-MIT-PathCreator
// SPDX-License-Identifier: MPL-2.0
```

6.2.2. *Using `REUSE.toml`.*

We used the `REUSE.toml` to license large amounts of code used from others (mostly libraries such as `LeanTween` and `BetterMinimal`).

Example

```
[[annotations]]
path = ["Tanks/Assets/LeanTween/**"]
precedence = "override"
SPDX-FileCopyrightText = ["2017 Russell Savage", "2001 Robert Penner"]
SPDX-License-Identifier = "LicenseRef-MIT-LeanTween"
```

7. Licensing Art

Tip

I highly recommend reading Creative Commons' FAQ page for more information

Tip

For learning about license compatibility between Creative Commons licenses, see this question in Creative Common's FAQ.

7.1. UPGRADE-Contributed Art.

Licensing original art created by UPGRADE for Catanks was very easy due to the license agreement. We simply used the `REUSE.toml` file to license all the art under the `CC BY-SA 4.0` license.

Example

```
[[annotations]]
path = [
  "Tanks/Assets/Models/**",
  "Tanks/Assets/Images/**",
  "AchievementsWithBorders/**",
  "Tanks/Assets/Audio/CustomSoundEffects/**",
  "Tanks/Assets/Audio/OfficialTracks/**"
  ]
precedence = "override"
SPDX-FileCopyrightText = "2024 The Catanks Contributors"
SPDX-License-Identifier = "CC-BY-SA-4.0"
```

7.2. Reused Art: CC0.

7.2.1. *No Modifications Made.*

To reuse art that is licensed under `CC0`, no attribution is needed. However, I did include links to all art used in the `README`. In order to make licensing easier, in the `REUSE.toml`, I marked the copyright holders as `NOASSERTION`.

Example

```
[[annotations]]
path = ["Tanks/Assets/Audio/OpenSourceSounds/CC0/**"]
precedence = "override"
SPDX-FileCopyrightText = "NOASSERTION"
SPDX-License-Identifier = "CC0-1.0"
```

7.2.2. *Modifications Made.*

If enough modifications are made to consider it a new work (the bar is fairly low, but it is more than simply cutting audio for example), the new work can be licensed under `CC BY-SA 4.0`.[13] It is courteous to include attribution or a link to the original work somewhere in the project (perhaps in the `README` as mentioned in the previous section).

7.3. Reused Art: CC BY 3.0 and 4.0.

7.3.1. *No Modifications Made.*

If no modifications are made, you **must** include the file in the `REUSE.toml` under the license it was released under. Additionally, there must be a note included that gives attribution. I used the SPDX-FileNotice tag for this.

[13] https://law.stackexchange.com/questions/33299/copyright-when-using-works-in-cc0-public-domain

Example

```
[[annotations]]
path="Tanks/Assets/Audio/OpenSourceSounds/Other/bounce.wav"
precedence = "override"
SPDX-FileCopyrightText = ["NoiseCollector"]
SPDX-License-Identifier = "CC-BY-3.0"
SPDX-FileNotice = "Audio by NoiseCollector -- https://freesound.org/s/111361/ --
License: Attribution 3.0"
```

7.3.2. *Modifications Made.*

7.3.2.1. *Original Licensed Under `CC BY 4.0`.*

If modifications are made to art originally licensed under `CC BY 4.0`, one can license the new work under `CC BY-SA 4.0`.[14, 15] However, attribution to the original author must be included in the `REUSE.toml`.

Example

```
[[annotations]]
path="Tanks/Assets/Audio/OpenSourceSounds/Other/mechanical-gadget.mp3"
precedence = "override"
SPDX-FileCopyrightText = ["2024 The Catanks Contributors", "kiddpark"]
SPDX-License-Identifier = "CC-BY-SA-4.0"
SPDX-FileNotice = "Audio modified by The Catank Contributors.\nOriginal audio made by
kiddpark, licensed under CC-BY-4.0: https://freesound.org/s/201159/"
```

7.3.2.2. *Original Licensed Under `CC BY 3.0`.*

For art originally licensed under `CC BY 3.0`, there are a couple options:

1. License final work under `CC BY 3.0`
2. License final work under `CC BY-SA 3.0`[14]

8. Reusing Fonts

8.1. **Open Font License (OFL).**

The OFL is one of the most popular open source font licenses. However, there are some caveats that one should keep in mind when using OFL licensed fonts.

8.1.1. *Reserved Names.*

Some OFL licensed fonts might have a “reserved name” which is a font name that those who modify the original font cannot use for their version. It is usually specified at the top of the OFL license text. In order to properly license OFL fonts with a reserved name using REUSE do the following:

1. Utilize the license with SPDX identifier `OFL-1.1-RFN`
2. Identify any reserved names in the `SPDX-FileNotice` tag in the `REUSE.toml`

[14] https://opensource.stackexchange.com/a/12364

[15] https://creativecommons.org/faq/#if-i-derive-or-adapt-material-offered-under-a-creative-commons-license-which-cc-licenses-can-i-use

Example

```
[[annotations]]
path = [
  "Tanks/Assets/TextMesh Pro/Fonts/LiberationSans.ttf",
  "Tanks/Assets/TextMesh Pro/Resources/Fonts & Materials/LiberationSans SDF -
Fallback.asset",
  "Tanks/Assets/TextMesh Pro/Resources/Fonts & Materials/LiberationSans SDF.asset",
  "Tanks/Assets/TextMesh Pro/Resources/Fonts & Materials/LiberationSans SDF - Drop
Shadow.mat",
  "Tanks/Assets/TextMesh Pro/Resources/Fonts & Materials/LiberationSans SDF -
Outline.mat"
  ]
precedence = "override"
SPDX-FileCopyrightText = ["2010 Google Corporation", "2012 Red Hat, Inc."]
SPDX-License-Identifier = "OFL-1.1-RFN"
SPDX-FileNotice = ["Reserved Font \"Arimo\", \"Tinos\", and \"Cousine\"", "Reserved Font
Name \"Liberation\""]
```

8.1.2. *No Reserved Names.*

If the font has no reserved names, use the license with the SPDX identifier `OFL-1.1-no-RFN`

8.1.3. `.asset` *File Derived From the Font.*

According to the OFL:

> The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.
>
> — Open Font License, https://openfontlicense.org/open-font-license-official-text/

To me, this would imply that the `.asset` and `.mat` files that TextMeshPro produces from the font in question must also be licensed under the OFL.

Example

```
[[annotations]]
path = [
    "Tanks/Assets/TextMesh Pro/Fonts/LiberationSans.ttf",
    "Tanks/Assets/TextMesh Pro/Resources/Fonts & Materials/LiberationSans SDF -
Fallback.asset",
    "Tanks/Assets/TextMesh Pro/Resources/Fonts & Materials/LiberationSans SDF.asset",
    "Tanks/Assets/TextMesh Pro/Resources/Fonts & Materials/LiberationSans SDF - Drop
Shadow.mat",
    "Tanks/Assets/TextMesh Pro/Resources/Fonts & Materials/LiberationSans SDF -
Outline.mat"
    ]
precedence = "override"
SPDX-FileCopyrightText = ["2010 Google Corporation", "2012 Red Hat, Inc."]
SPDX-License-Identifier = "OFL-1.1-RFN"
SPDX-FileNotice = ["Reserved Font \"Arimo\", \"Tinos\", and \"Cousine\"", "Reserved Font
Name \"Liberation\""]
```

9. Licensing Gotchas

9.1. TextMesh Pro.

9.1.1. *Unity Companion License.*

TextMesh Pro is licensed under the Unity Companion license. For the most part, the files that one would include in a git repository do not fall under TextMesh Pro's license. However some files do. Particularly the following files:

- All non-custom TextMesh Pro shaders
- `LineBreaking Leading Characters.txt` and `LineBreaking Following Characters.txt`
- `Default Style Sheet.asset`

Example

```
[[annotations]]
path=[
  "Tanks/Assets/TextMesh Pro/Resources/LineBreaking Leading Characters.txt",
  "Tanks/Assets/TextMesh Pro/Resources/LineBreaking Following Characters.txt",
  "Tanks/Assets/TextMesh Pro/Shaders/*",
  "Tanks/Assets/TextMesh Pro/Resources/Style Sheets/Default Style Sheet.asset"
  ]
precedence = "override"
SPDX-FileCopyrightText = "2022 Unity Technologies ApS"
SPDX-License-Identifier = "LicenseRef-Unity-Companion"
```

9.1.2. *Files to Exclude.*

I recommend excluding the TextMesh Pro documentation from your git repository (as it is uneeded). Additionally, I recommend excluding the EmojiOne files from the TextMesh Pro files as they have a weird license. See **Appendix B** for example additions to your `.gitignore` file.

9.2. Licensing Info in Built Game.

Many open source licenses (such as MIT and BSD licenses for example) require the copyright text also be included with the compiled program. To that end, ensure that a file (in Catanks we called this file `THIRD-PARTY-LICENSES.md`) is included with the compiled program and contains the text for the licenses used. For an example, see Catanks' `THIRD-PARTY-LICENSES.md`.

> 🔥 **Tip**
>
> You do not need to include the Creative Commons attributions in this file. Instead, you can link to a page that contains the attributions[16] (such as the REUSE.toml on your git repository).

9.3. `UIElementsSchema` Directory.

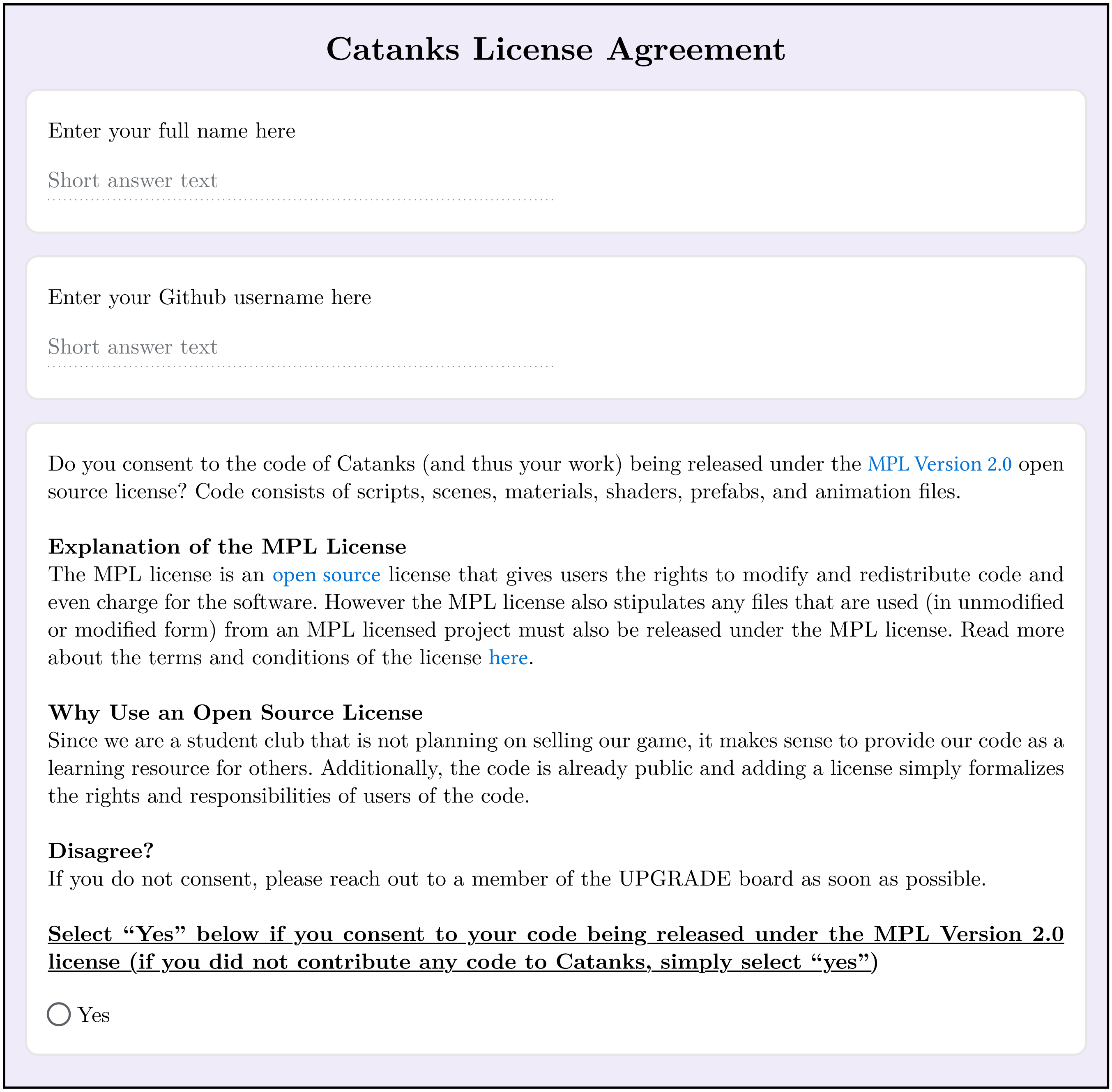
If you didn't add any files to the `UIElementsSchema` Directory, one can safely gitignore it.[17]

[16] https://creativecommons.org/faq/#how-do-i-properly-attribute-material-offered-under-a-creative-commons-license

[17] https://discussions.unity.com/t/uielementsschema/805120

Appendix A: Example License Agreement

For Catanks, we made the license agreement on Google Forms.



Catanks License Agreement

Enter your full name here

Short answer text

Enter your Github username here

Short answer text

Do you consent to the code of Catanks (and thus your work) being released under the MPL Version 2.0 open source license? Code consists of scripts, scenes, materials, shaders, prefabs, and animation files.

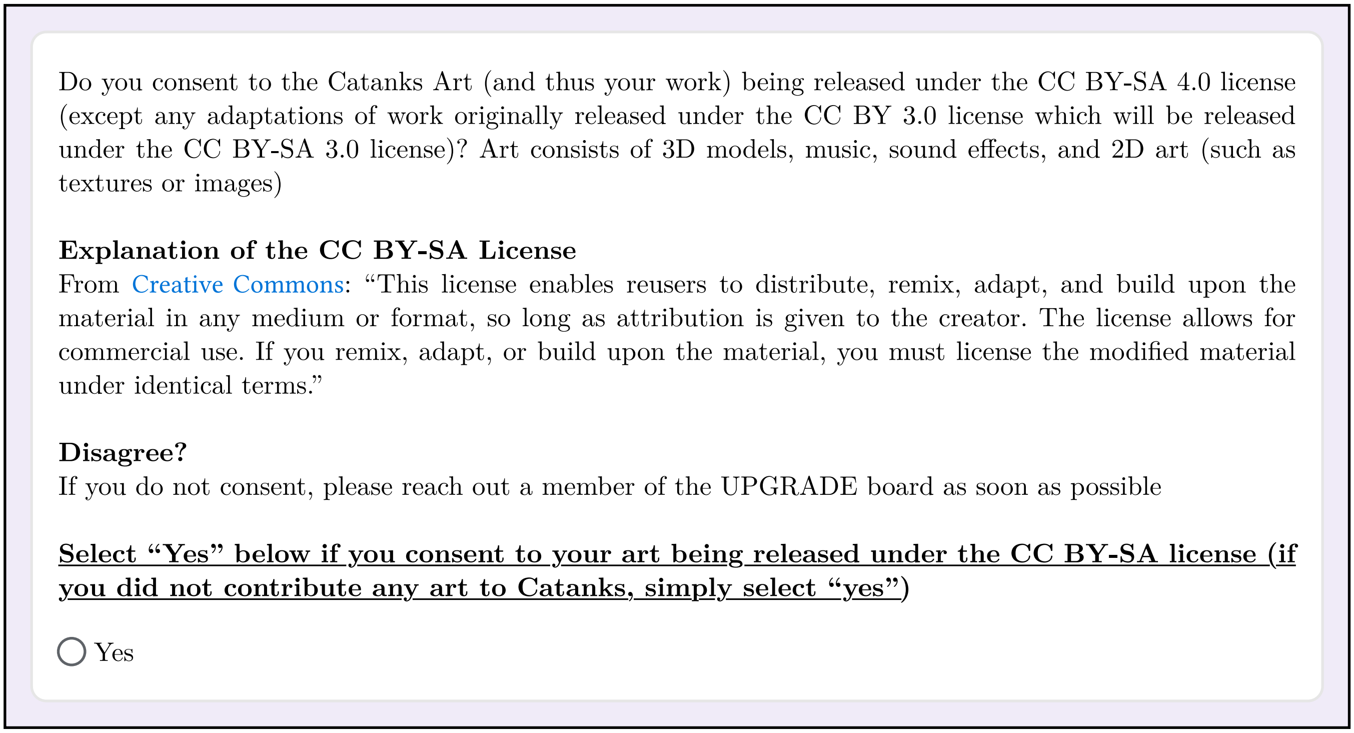
Explanation of the MPL License
The MPL license is an open source license that gives users the rights to modify and redistribute code and even charge for the software. However the MPL license also stipulates any files that are used (in unmodified or modified form) from an MPL licensed project must also be released under the MPL license. Read more about the terms and conditions of the license here.

Why Use an Open Source License
Since we are a student club that is not planning on selling our game, it makes sense to provide our code as a learning resource for others. Additionally, the code is already public and adding a license simply formalizes the rights and responsibilities of users of the code.

Disagree?
If you do not consent, please reach out to a member of the UPGRADE board as soon as possible.

Select "Yes" below if you consent to your code being released under the MPL Version 2.0 license (if you did not contribute any code to Catanks, simply select "yes")

○ Yes



Do you consent to the Catanks Art (and thus your work) being released under the CC BY-SA 4.0 license (except any adaptations of work originally released under the CC BY 3.0 license which will be released under the CC BY-SA 3.0 license)? Art consists of 3D models, music, sound effects, and 2D art (such as textures or images)

Explanation of the CC BY-SA License
From Creative Commons: “This license enables reusers to distribute, remix, adapt, and build upon the material in any medium or format, so long as attribution is given to the creator. The license allows for commercial use. If you remix, adapt, or build upon the material, you must license the modified material under identical terms.”

Disagree?
If you do not consent, please reach out a member of the UPGRADE board as soon as possible

Select “Yes” below if you consent to your art being released under the CC BY-SA license (if you did not contribute any art to Catanks, simply select “yes”)

○ Yes

Appendix B: Example `.gitignore`

```
# DS_Store
**/.DS_Store

# TextMesh Pro documentation
Assets/TextMesh Pro/Documentation/

# EmojiOne
Assets/TextMesh Pro/**/EmojiOne*

UIElementsSchema
```

UPGRADE, Philadelphia, PA, 19104