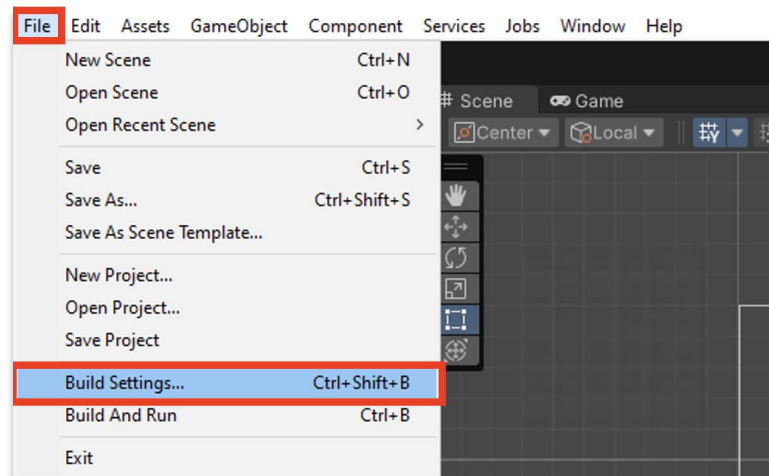


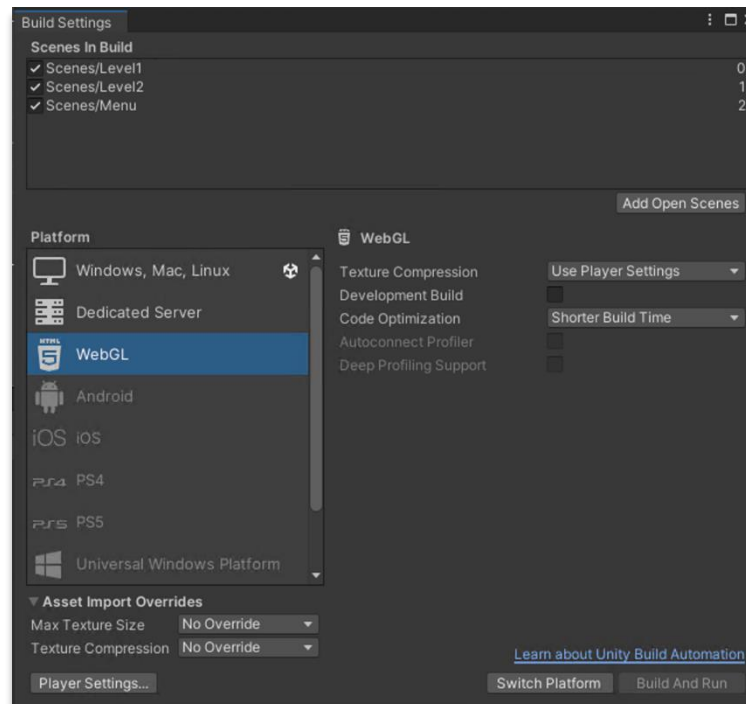


Creating a WebGL link

1 In Unity, click File, then Build Settings.



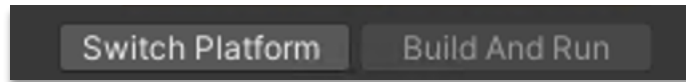
2 Make sure all scenes are checked. Under Platform, click WebGL. You may need to install it from the Unity Hub.





Creating a WebGL link

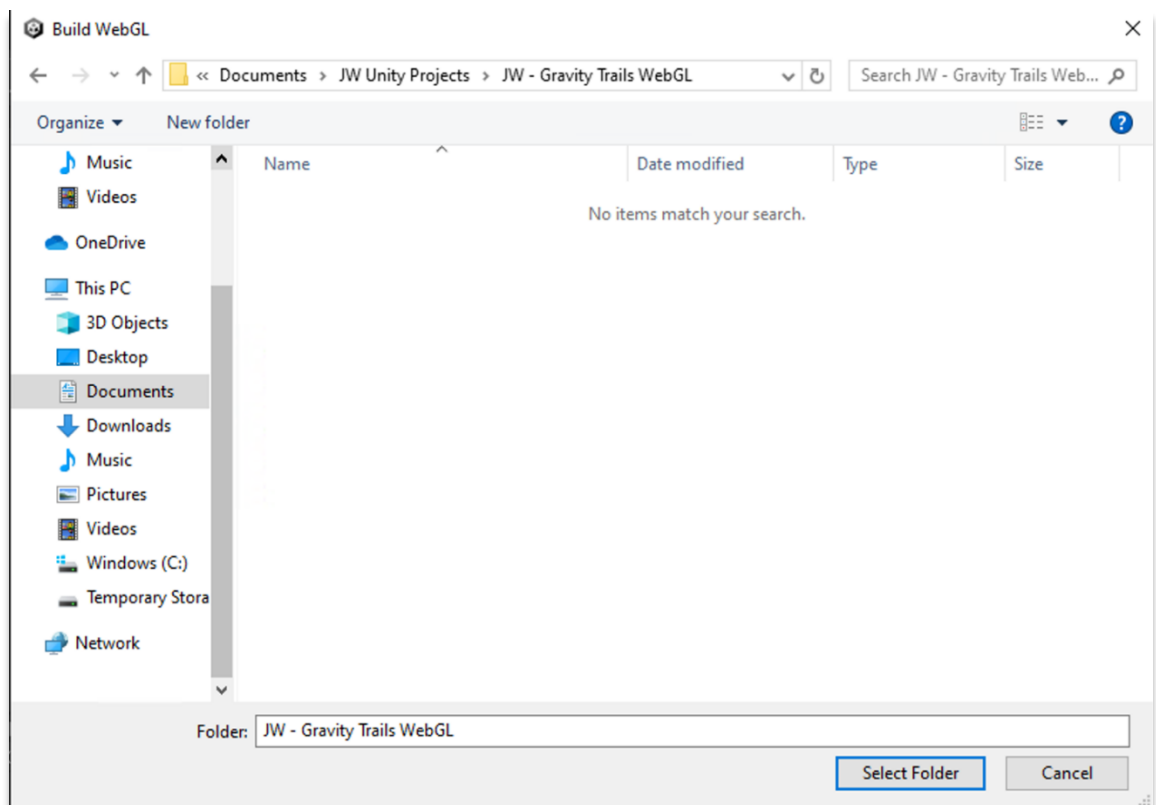
- 3 Click "Switch Platform." This may take a few minutes.



- 4 Click Build And Run.



- 5 Create a folder and save the WebGL files in a location you will remember.



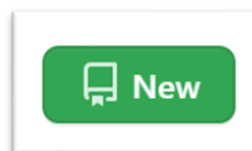


Creating a WebGL link

- 6 Go to github.com/login and sign in. If you do not already have an account, create a free account.

A screenshot of the GitHub login page. At the top is the GitHub logo and the text "Sign in to GitHub". Below this is a form with two input fields: "Username or email address" and "Password". There is a "Forgot password?" link next to the password field. A green "Sign in" button is below the fields. Below the button is a link "Sign in with a passkey" and another link "New to GitHub? Create an account". At the bottom of the page are links for "Terms", "Privacy", "Docs", "Contact GitHub Support", "Manage cookies", and "Do not share my personal information".

- 7 In your dashboard, click the "New" button next to Repositories.





Creating a WebGL link

8 Create a new repository.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner *

Repository name *

jonahwagner

 /

JW Gravity Trails

✓ Your new repository will be created as JW-Gravity-Trails.

The repository name can only contain ASCII letters, digits, and the characters -, ., and _.

Great repository names are short and memorable. Need inspiration?

Description (optional)

☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Initialize this repository with:

☒ Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

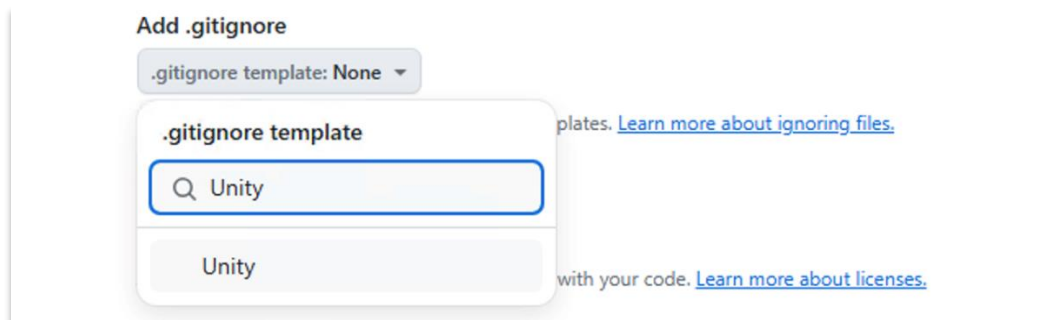
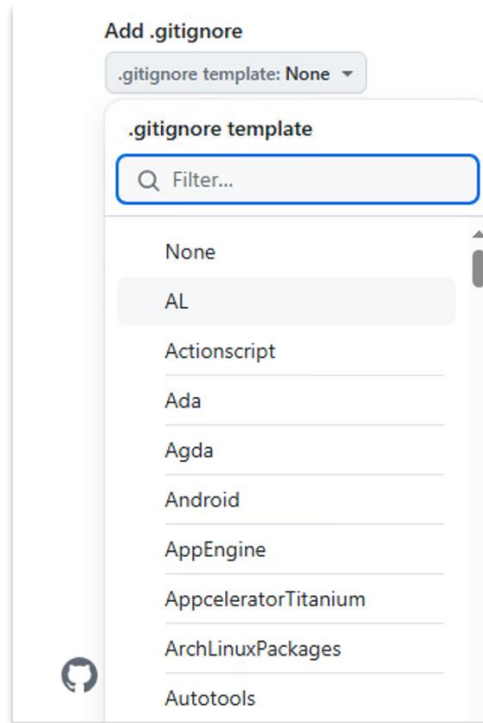
Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Ensure the box is checked to add a README file.



Creating a WebGL link

- 9 Click the “.gitignore template: None” dropdown menu, and type “Unity”.





Creating a WebGL link

- 10** Leave the “Add a license” box as none, unless you know what kind of license you want to use.

Choose a license

License: **None** ▼

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set **main** as the default branch. Change the default name in your [settings](#).

- 11** Click the Create repository button.

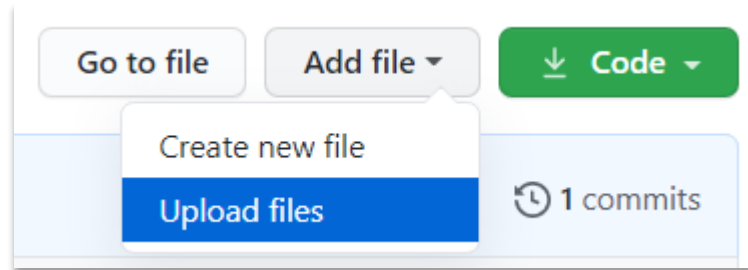
Create repository

The screenshot shows a GitHub repository page for 'jonahwagner / JW-Gravity-Trails'. The repository is public and has 1 branch (main) and 0 tags. It contains 1 commit (76f614c) and 2 files: .gitignore and README.md. The README file is open, showing the title 'JW-Gravity-Trails'. The right sidebar shows the 'About' section with no description, website, or topics provided. It also shows 'Releases' and 'Packages' sections, both with no published items. The footer of the page shows the GitHub logo and copyright information for 2024.

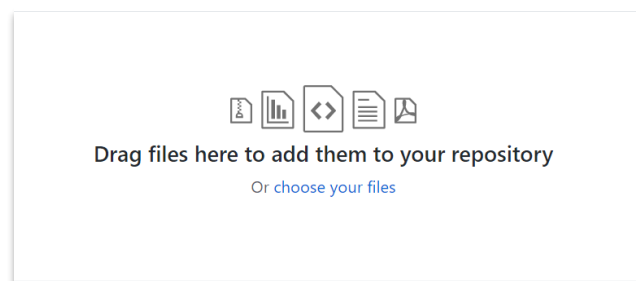


Creating a WebGL link

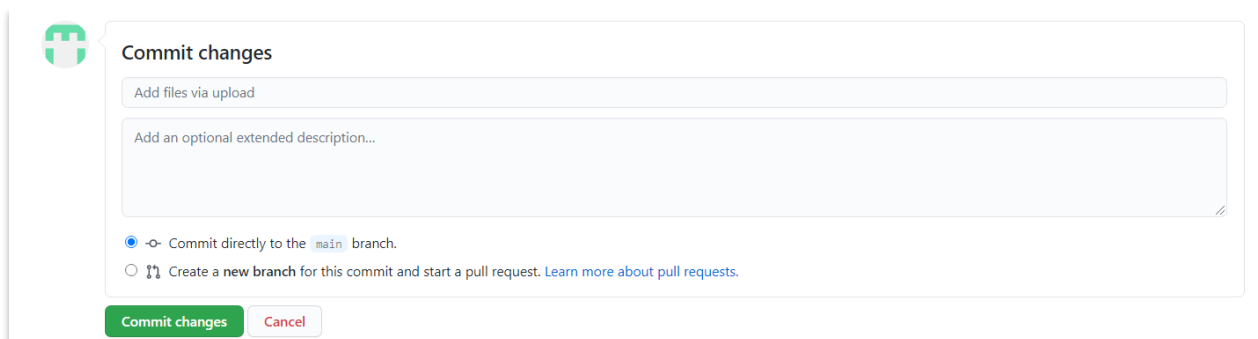
12 Click Add file, then Upload files.



13 Click "choose your files," then locate the folder where you saved your project's WebGL files. Upload *index.html*, *Build/*, and *TemplateData/*



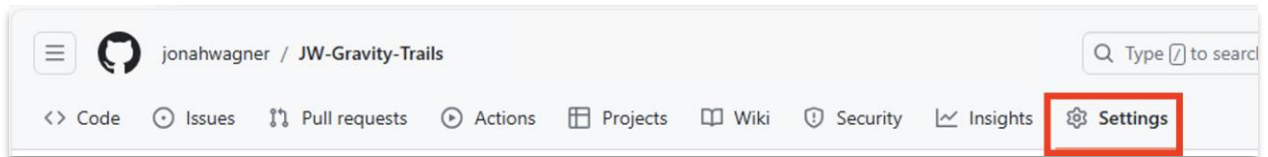
14 Click Commit changes.



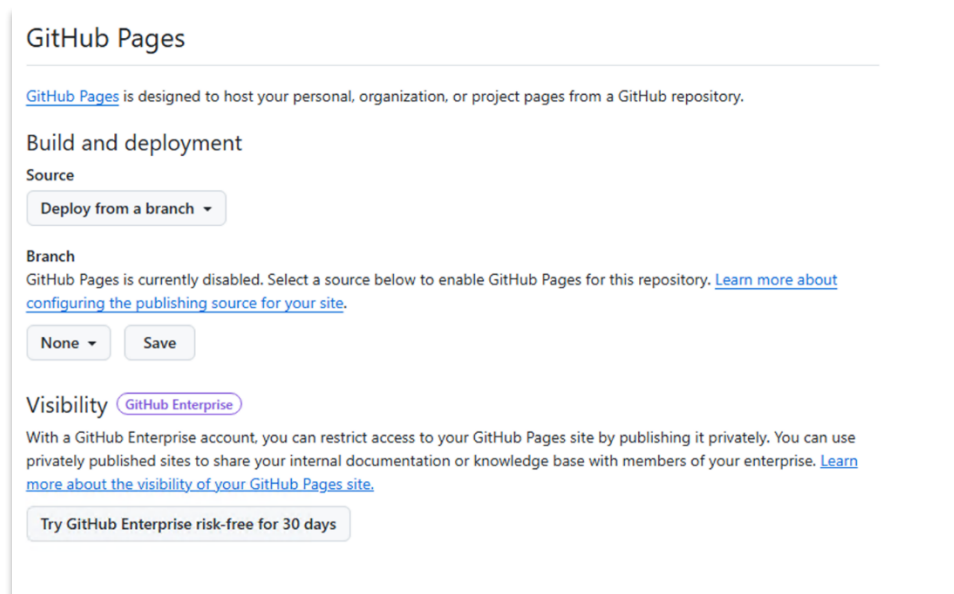


Creating a WebGL link

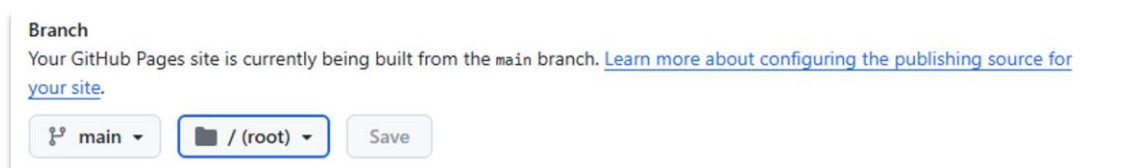
15 Once your files are uploaded, click on "Settings".



16 On the left select Pages.



17 Make sure your Branch settings match this image.



Wait while the page is being published, you may have to click refresh until the page is published.



Creating a WebGL link

- 18** Click on the hyperlink to open your project page. Test out your project to make sure all parts work and look the way you want.

GitHub Pages

Use this link to share your project!

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://jonahwagner.github.io/JW-Gravity-Trails/>
Last **deployed** by [jonahwagner](#) 1 minute ago

Visit site

Click on the blue button to expand the game to full screen.

A screenshot of the 'Gravity Trails' game. The title 'Gravity Trails' is at the top. Below it is a green 'Start Game' button. Underneath is a control panel titled 'Avatar Controls' with buttons for 'Move Left', 'Move Right', 'Gravity Flip', 'Throw', 'Space', and 'Left CTRL'. The game background shows a 2D platformer scene with floating islands, trees, and mountains. The Unity WebGL logo is in the bottom left, and 'PS- Gravity Trails' is in the bottom right.

- 19** Use this GitHub URL to share your completed project!



Creating a WebGL link

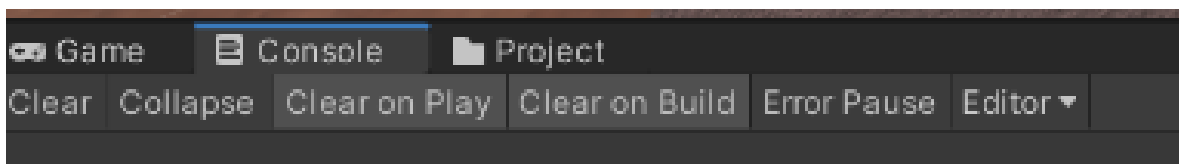
How to Reduce Game Size in Unity Tips and Tricks

Oh no! Your Unity project is too large to be loaded into GitHub. Try the tips below to reduce the size of your Unity project then rebuild the project and upload the new build files to GitHub.

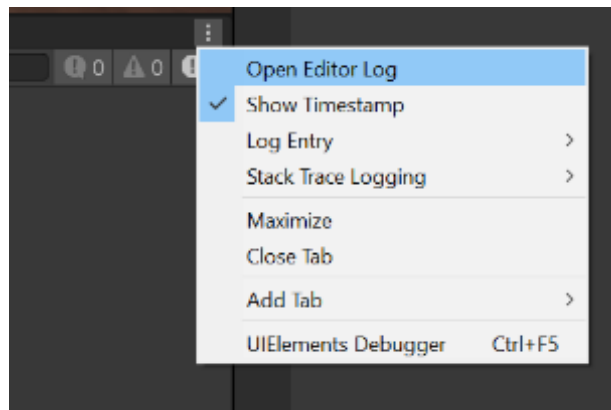
Note: GitHub allows file uploading up to 25MB only. If you try uploading files larger than 25 MB, you will see the error message, *'Yowza, that's a big file. Try again with a file smaller than 25MB'*.

First, you will want to determine which Assets in your project take up the most space in the build. This information can be found in the **Editor Log** after you have performed the build.

Go to the **Console window**:



Then click the small drop-down panel in the top right and select **Open Editor Log**.





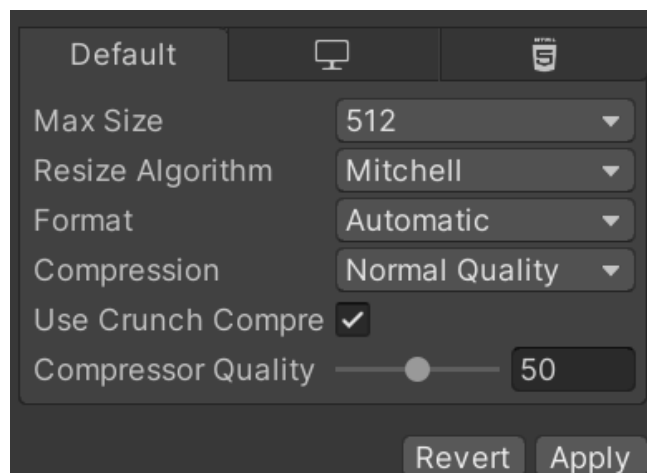
Creating a WebGL link

The **Editor Log** provides a summary of Assets broken down by type and lists them in order of size contribution.

```
-----
Build Report
Uncompressed usage by category (Percentages based on user generated assets only):
Textures          67.1 mb  53.9%
Meshes            25.6 mb  20.6%
Animations        3.4 kb   0.0%
Sounds            3.1 mb   2.5%
Shaders           4.9 mb   4.0%
Other Assets      1.1 mb   0.9%
Levels            8.2 mb   6.6%
Scripts           1.0 mb   0.8%
Included DLLs     13.3 mb  10.7%
File headers      52.5 kb   0.0%
Total User Assets 124.3 mb 100.0%
Complete build size 189.2 mb

Used Assets and files from the Resources folder, sorted by uncompressed size:
18.9 mb      10.0% Assets/Resources/Assets/MMSEV/MMSEV.obj
8.0 mb      4.2% Assets/Resources/Samples/Terrain Tools/0.1.0-preview/Terrain Assets/BrushTextures/errosion02.tif
8.0 mb      4.2% Assets/Resources/Samples/Terrain Tools/0.1.0-preview/Terrain Assets/BrushTextures/errosion01.tif
5.0 mb      2.6% Resources/unity_builtin_extra
2.7 mb      1.4% Assets/Resources/Fonts/Controls.png
2.5 mb      1.3% Assets/Resources/Assets/viking_carbajal/viking_carbajal.obj
```

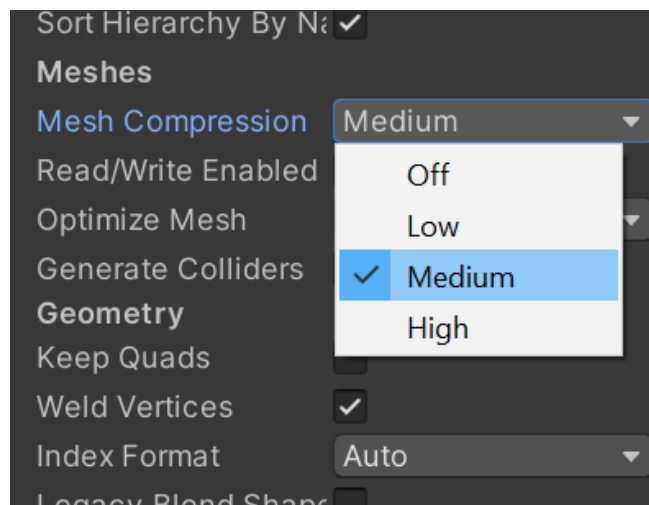
Textures usually take up the most space. Use **Compressed Texture formats** and **reduce the physical size** (in pixels) of Texture images. To do this, select the Texture, then in the Inspector window **decrease the Max Size** and choose to **Use Crunch Compre**. Then click **Apply** to see how it impacts the project. Return to the Inspector window to adjust the Max Size value as needed.





Creating a WebGL link

Meshes also take up a lot of file space. Enable **Mesh Compression** so that meshes and imported animation clips take up less space in your game file. To do this, select the mesh, then in the Inspector window set the **Mesh Compression** to **Low**, **Medium**, or **High**. See how it impacts the project, then return to the Inspector window to adjust the Mesh Compression value as needed.



For an in-depth walkthrough on how to do the steps listed above, follow along with this YouTube video: <https://youtu.be/7O21c8BzEzM>

Note: Unity strips most unused Assets during the build, so you don't gain anything by manually removing Assets from the Project. The only Assets that are not removed are scripts (which are generally very small anyway) and Assets in the Resources folder (because Unity can't determine which of these are needed and which are not). You should make sure that the only Assets in the Resources folder are the ones you need for the game.

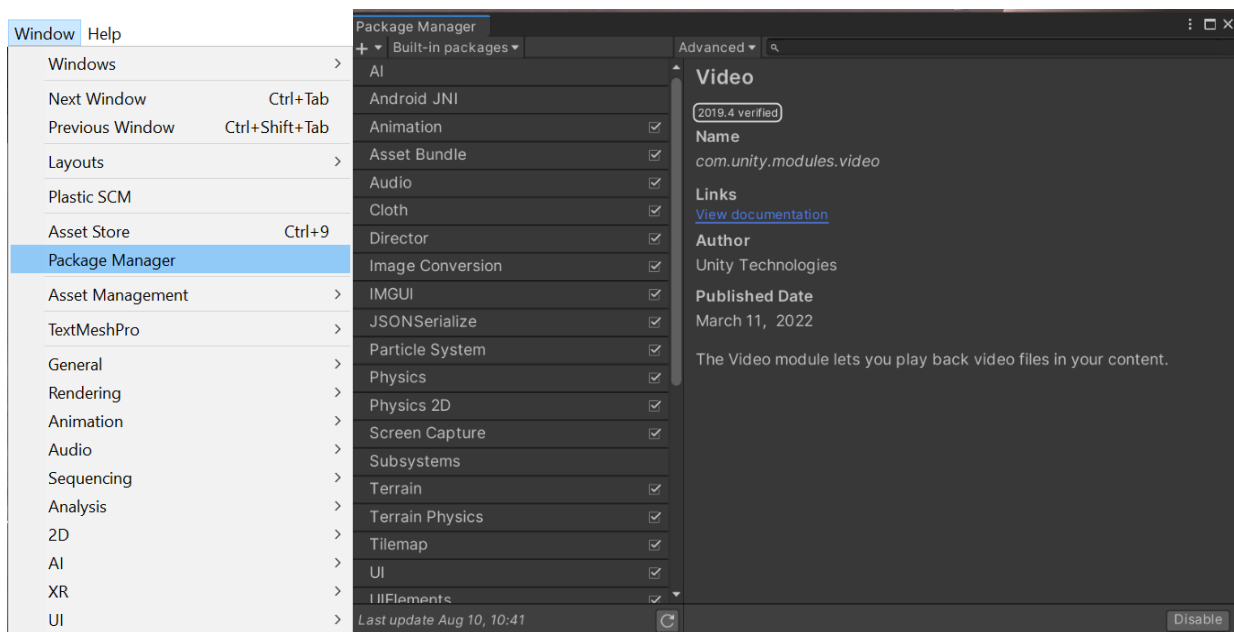


Creating a WebGL link

Other ways to reduce game size in Unity include:

Removing Unused Built-in Packages

When you create your project in Unity, the engine might import extra packages that might be of no use to you. Removing the packages might lessen your build size. To remove them, open the package manager from **Window > Package Manager > Built-In**



Remove Unused Scenes from Build Settings

Make sure you are building your project with only the necessary scenes.

