CODE NINJAS®

Gold Belt Ninja Guide

Activity 05: Release Candidate Phase

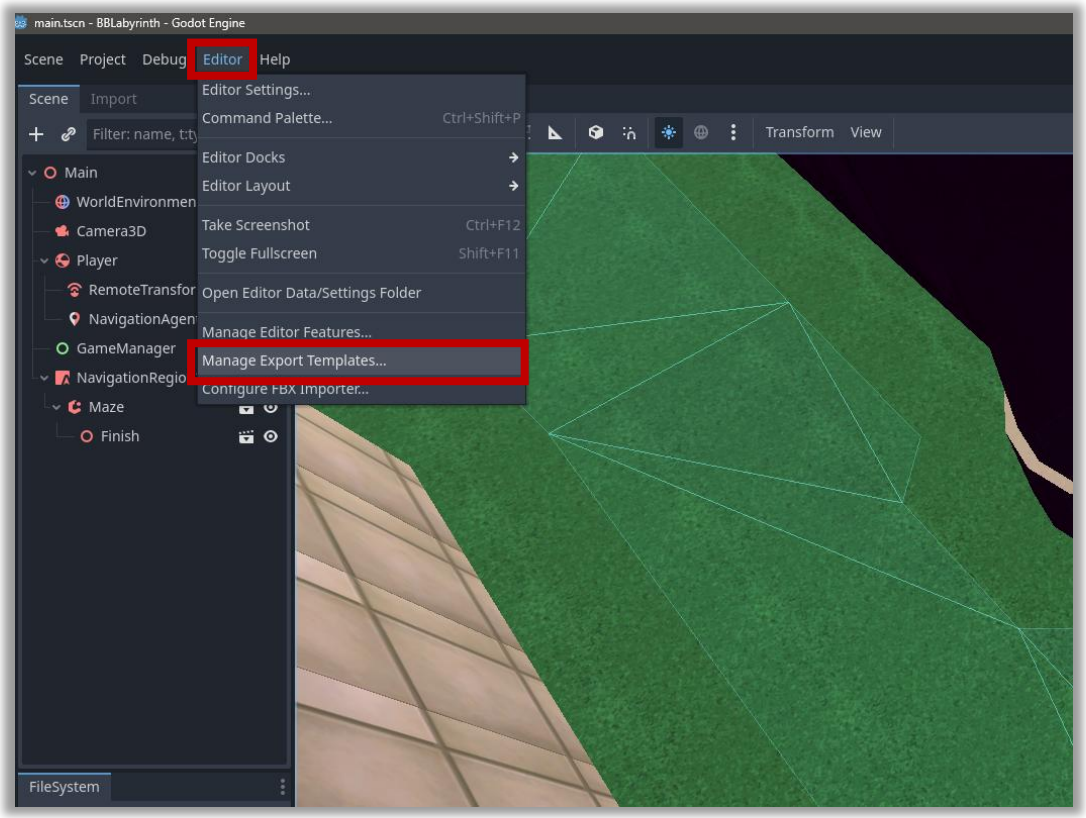
Release Candidate Phase

This is the finish line! For this phase, you need to build your project for the browser using WebGL. After building and hosting the project, complete another round of playtesting to ensure there are no bugs in the browser version.

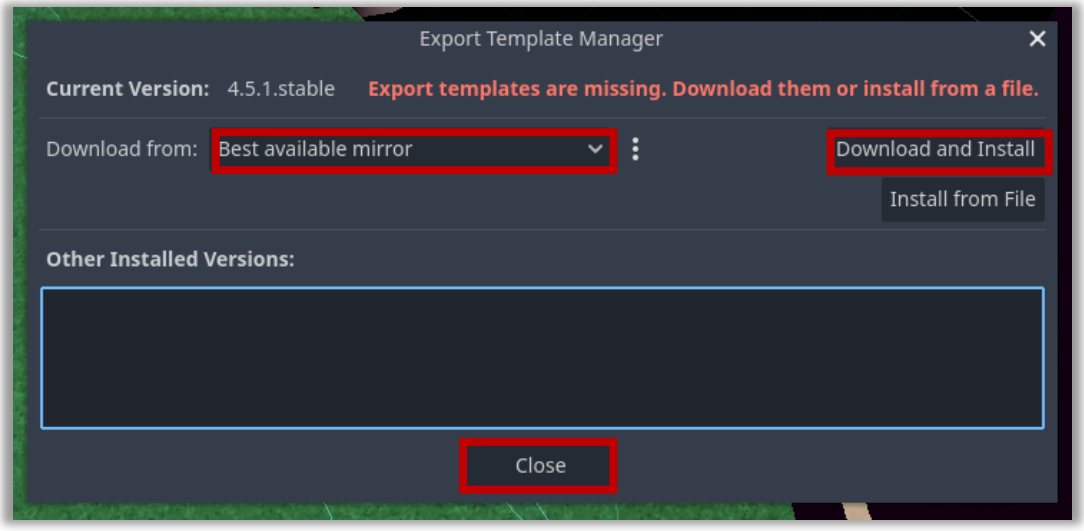
You should also complete a self – assessment of your project to make sure it meets Gold Belt project expectations. Once completed, your Code Sensei will help you submit your project for review from the Home Office EDU team!

Exporting your Godot Project as a WebGL Link

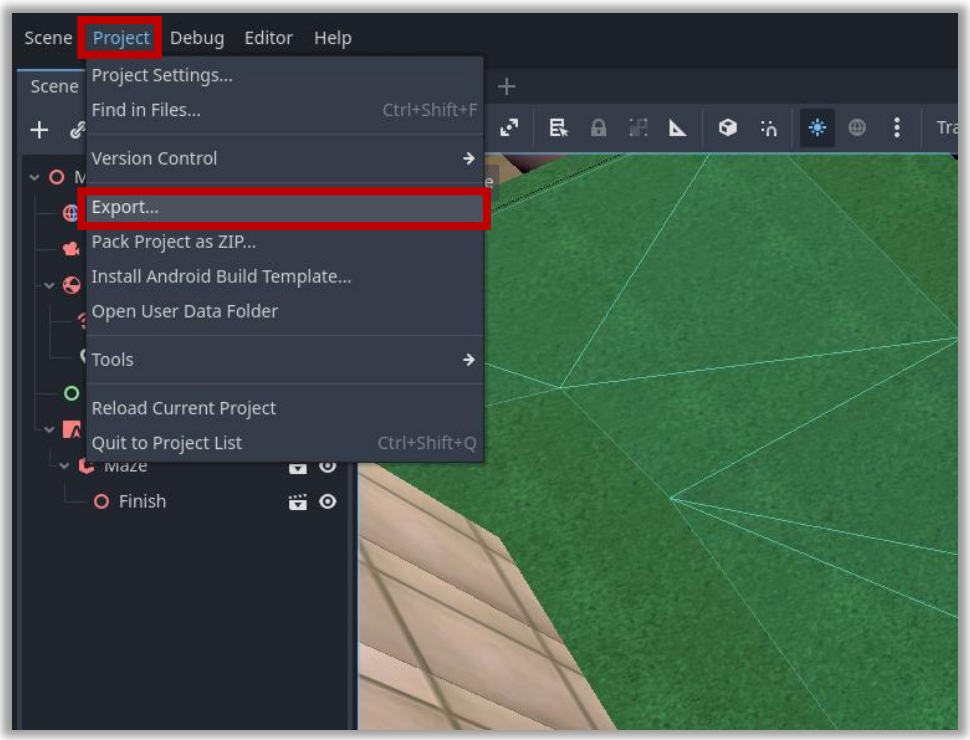
1 In Godot, click Editor > **Manage Export Templates**.



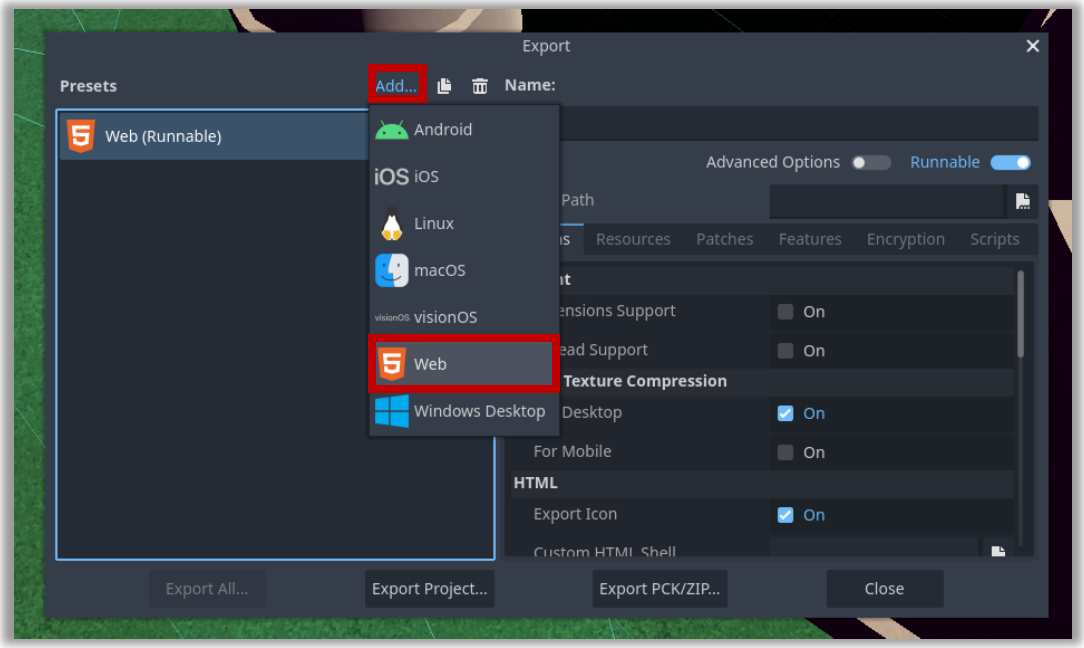
2 Choose **Best Available Mirror** from the dropdown. Click **Download and Install**. Wait for it to finish installing, then click **Close**.



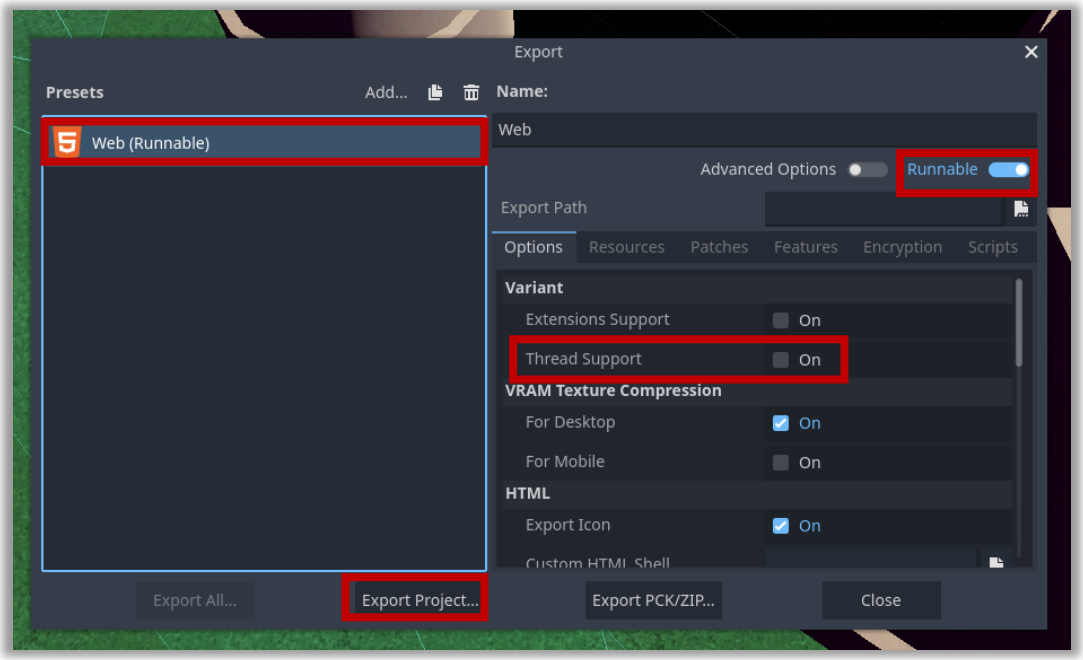
3 Click Project > **Export**.



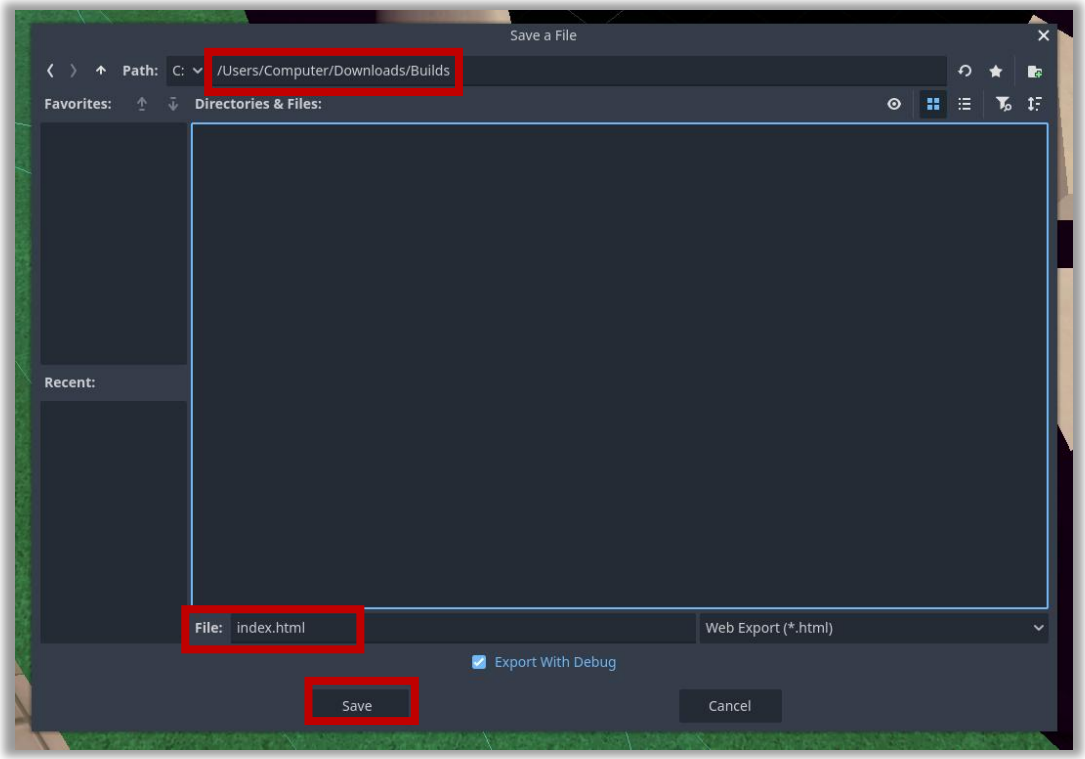
4 Click **Add** > **Web**.



5 Ensure **Runnable** is turned **on** and **Thread Support** is turned **off**. Click **Export Project**.

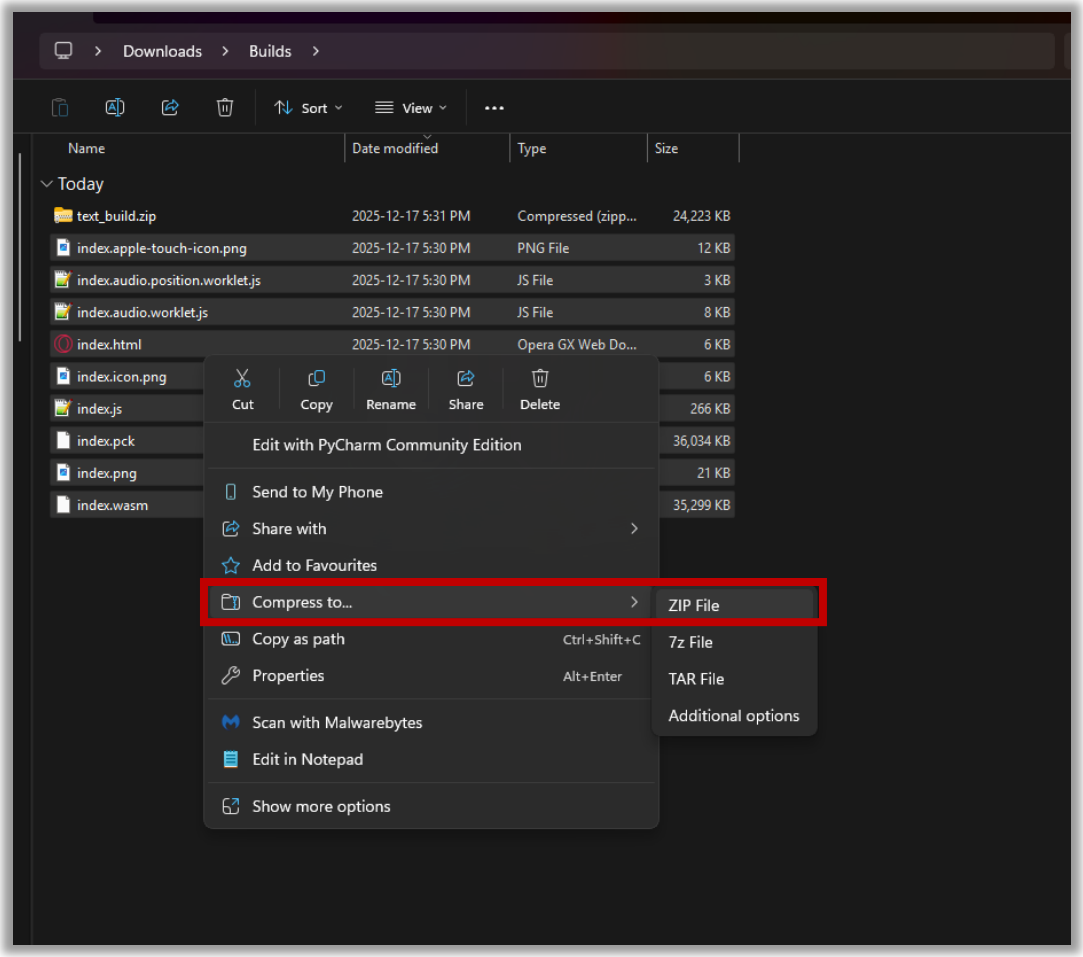


6 Create a new **Builds** folder. Save your project to this folder as **index.html**.



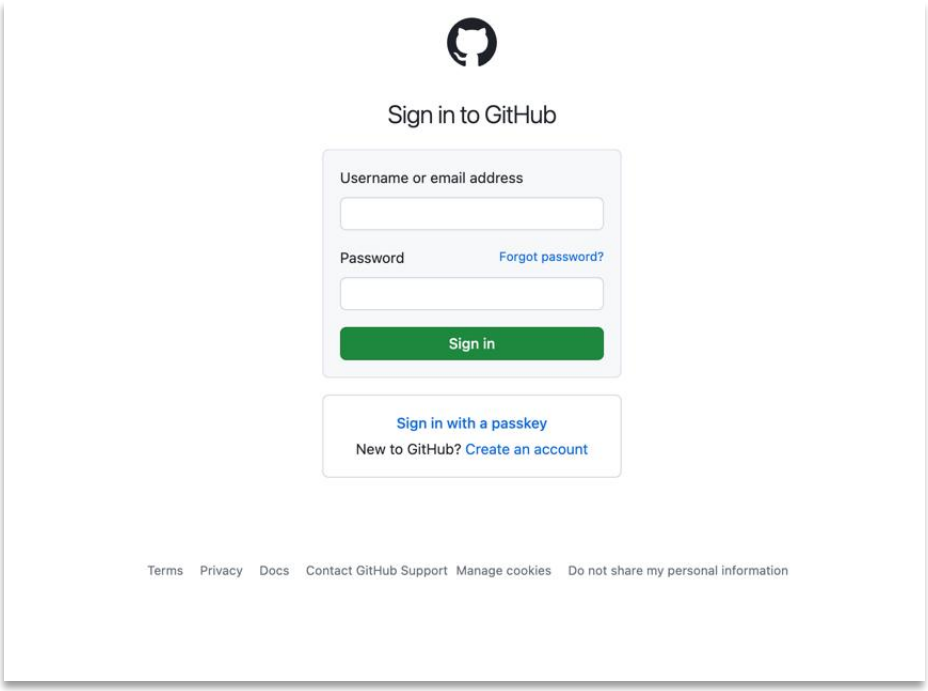
7 For certain websites like **itch.io**, the build must be compressed to a **zip folder** for security reasons.

If this is the case for you, select all the files in the build folder. Right click and select **Compress to...** > **ZIP File**. This will compress the selected files to a zip folder which can be uploaded online.

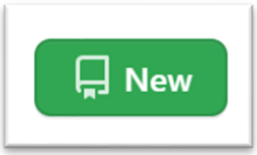


Congratulations, your project is now WebGL ready!

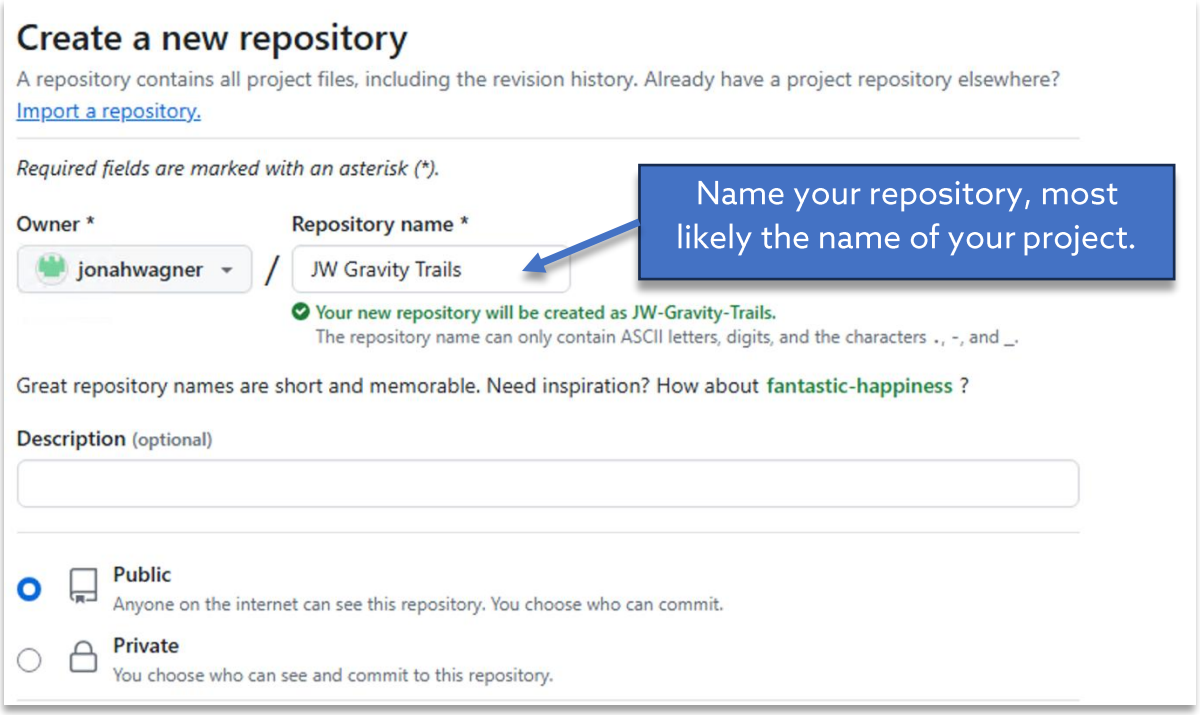
8 Go to github.com/login and sign in. If you do not already have an account, create a free account.

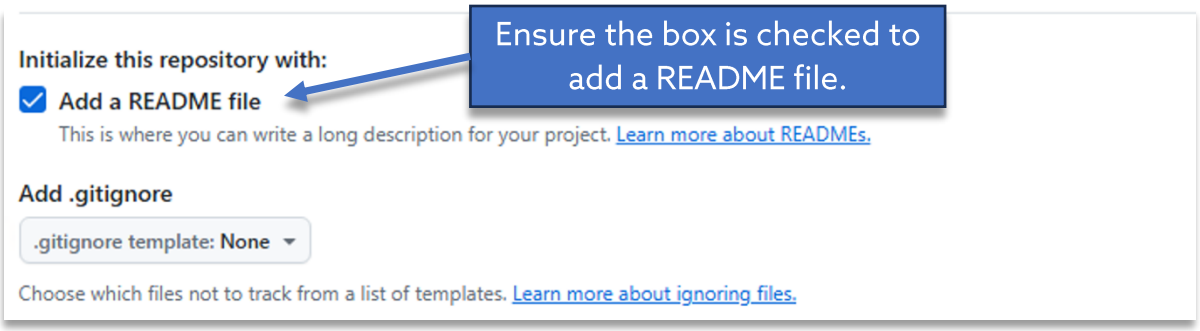


9 In your dashboard, click the "**New**" button next to Repositories.

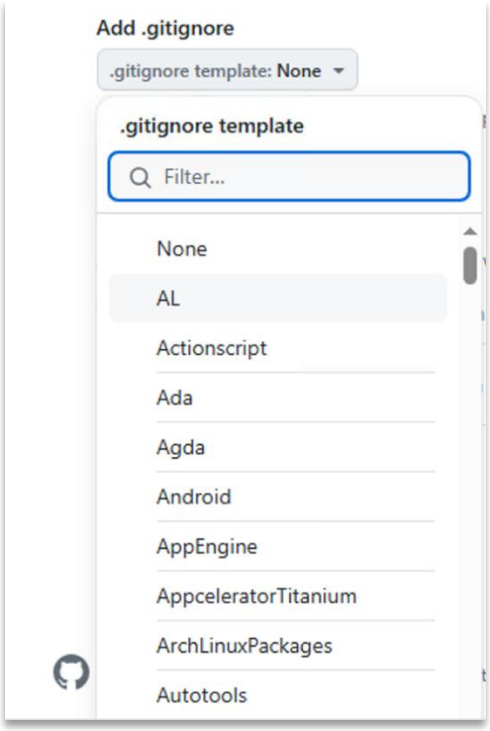


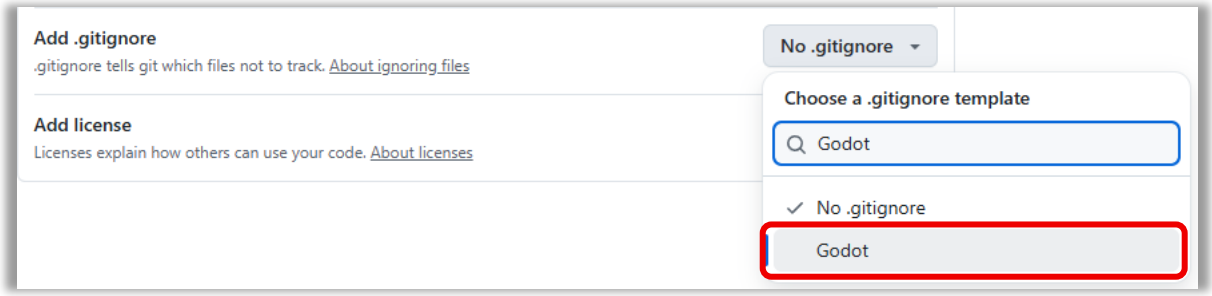
10 Create a new repository.



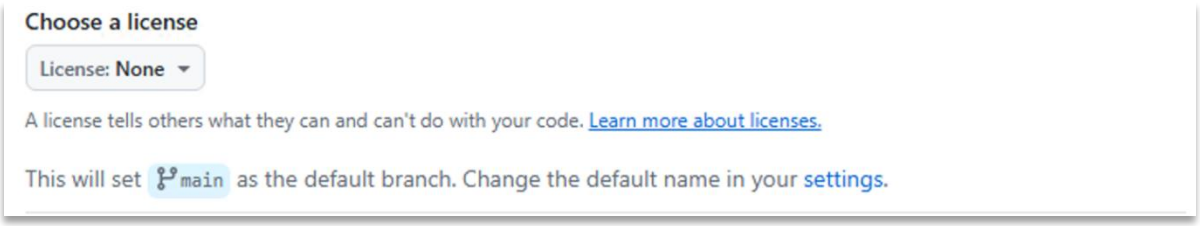


11 Click the **".gitignore template: None"** dropdown menu, and type "**Godot**".

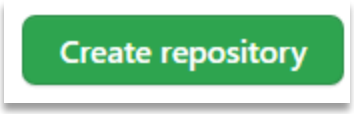


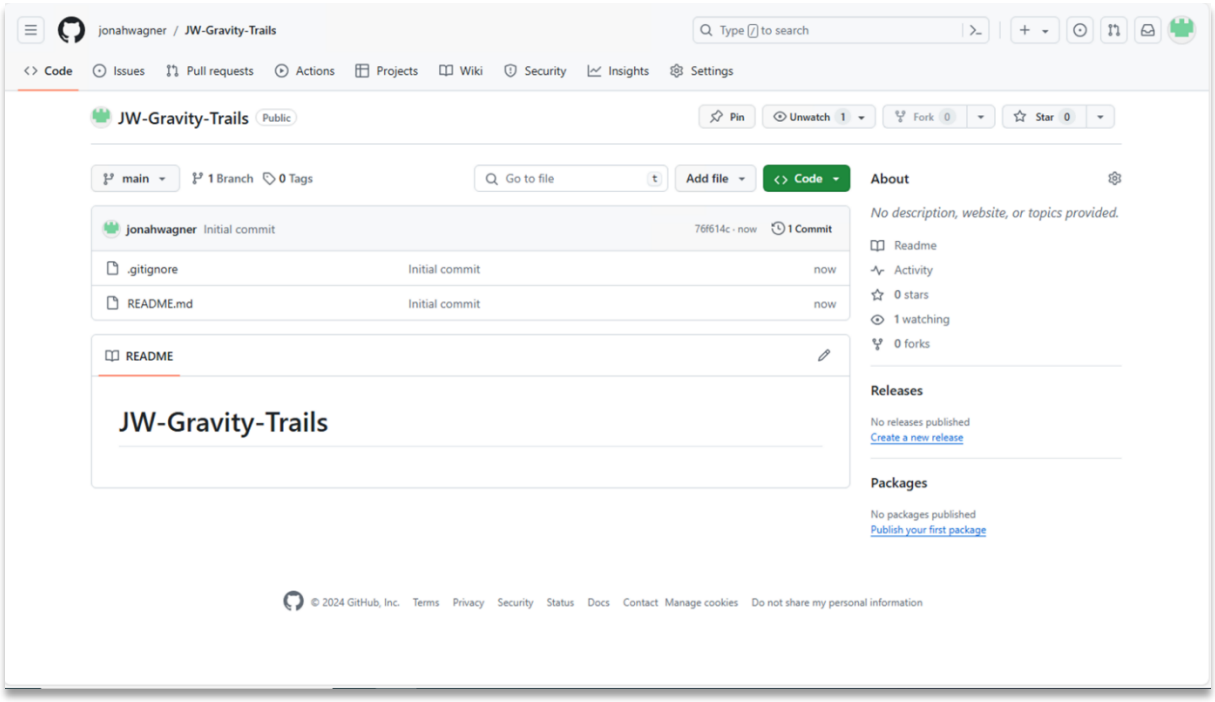


12 Leave the "**Add a license**" box as none, unless you know what kind of license you want to use.

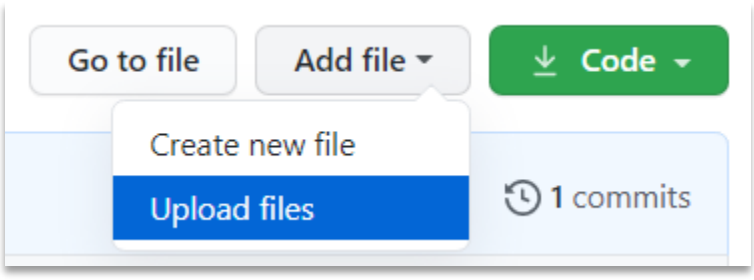


13 Click the **Create repository** button.

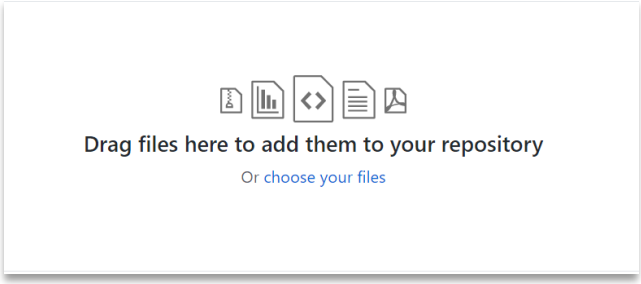




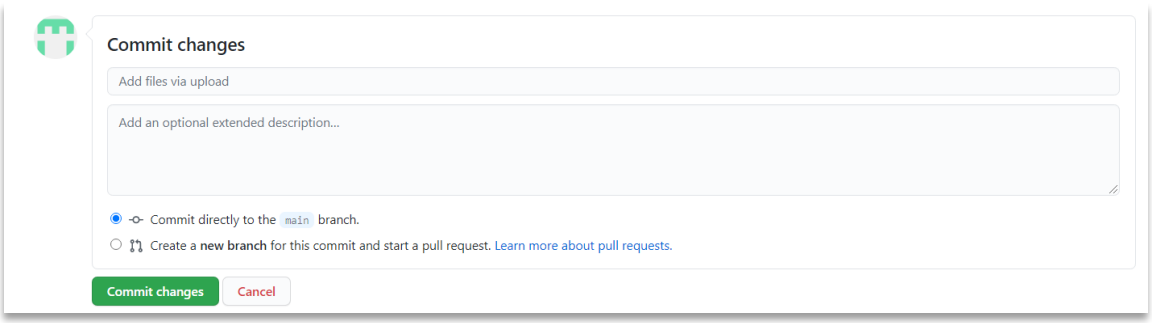
14 Click **Add file**, then **Upload files**.



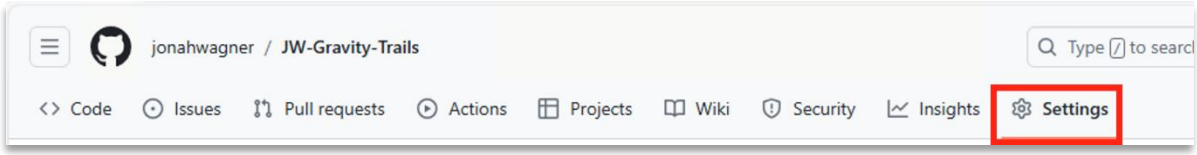
15 Click "**choose your files**," then locate the **Builds** folder from Step 6 where you saved your project's WebGL files. Upload the entire folder, including the **index.html** file.



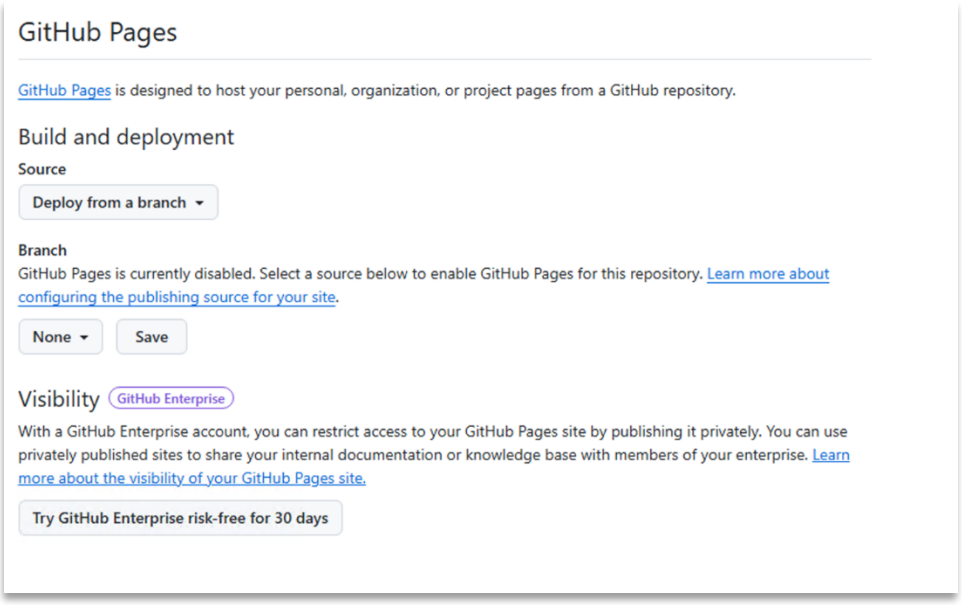
16 Click **Commit changes**.



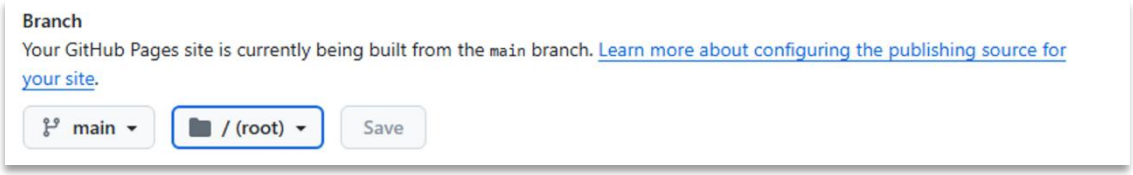
17 Once your files are uploaded, click on "Settings".



18 On the left, select **Pages**.

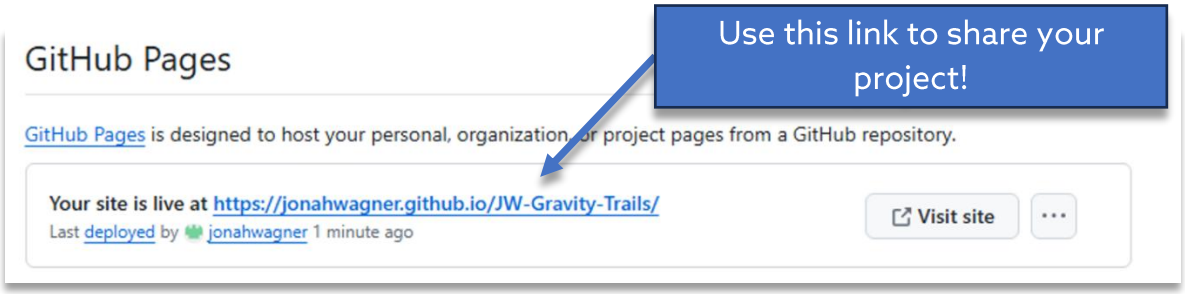


19 Make sure your **Branch settings** match this image.



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

20 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.



21 Use this **GitHub URL** to share your completed project!

Pause for **Sensei Stop #1**!

Have your Code Sensei assist you in hosting your WebGL project.

Afterwards, complete another round of playtesting using the WebGL build

Ninja Planning Documents

Complete the **Release Candidate Phase – Playtesting** portion of your Gold Belt Ninja Planning Document to find any bugs with the WebGL version.

Self-Assessment

Use the table below to confirm that your project meets the Gold Belt project criteria. Your project should align with the descriptions in the **Knight** column, before submitting to the Gold Belt committee for review.

Category	Sub Category	Recruit	Squire	Knight	Legend
Gameplay	**Difficulty**	Project is either extremely easy or difficult to play.	Project is appropriately difficult in most sections, but there are spontaneous spikes in difficulty for the player.	Project is appropriately difficult and allows for progression for the player.	Difficulty feels extremely rewarding and is appropriate for multiple playstyles. Multiple difficulty options may be present.
	Variety	Project contains limited variety, repetitive tasks, and few choices. There is a lack of visual diversity in environments and uniform challenges.	Game has somewhat more varied gameplay, but integration is lacking, with moderate environments, predictable challenges, and minimal player impact.	Game provides a rich variety in activities, immersive and visually captivating environments, varied challenges that demand adaptability, and meaningful player choices.	Players encounter impressive variety, stunning environments, dynamic challenges, and profound player impact through choices shaping the experience.
	Level Design	Lacks creativity and aesthetics; uses repetitive, linear structures with inconsistent pacing.	Designs offer basic engagement, moderate variety, limited exploration, and inconsistent pacing.	Users engage with creative layouts, diverse structures, player exploration, and well-balanced pacing.	Excels with thoughtful creativity, diverse structures, open exploration, and perfectly tuned pacing.

Art & Design	**Sprites & Models**	Visuals lack detail and customization. Textures are low-quality, and there's inconsistency in style.	Project uses basic visuals with moderate quality textures and models. There's room for more engaging and consistent art elements.	Sprites and models are engaging, featuring high-quality textures. A consistent style is maintained throughout the project.	Expertly designed visuals with top-tier textures and models. Project has a unique and stunning style.
	Music & Sound	Sound effects and music are missing or underutilized.	Project includes functional but unexceptional music and sound effects that could benefit from more diversity and better integration into the game world.	The music and sound effects are well-crafted, enhancing the atmosphere and gameplay experience with an engaging and immersive auditory backdrop.	Provide a masterful auditory experience, with music and sound effects that are not only perfectly suited to the game but elevate it to a new level of immersion and emotional impact.
	Animations	Animations are missing or out of place.	Animations are stiff, not exciting, and lack expressiveness, detracting from immersion.	Animations are fluid, expressive, and seamlessly transitioned, enhancing immersion and engagement	Animations are exceptionally fluid, deeply expressive, and seamlessly transitioned.

User Interface & Experience	**Menus Present**	Project does not contain menus and starts directly into gameplay.	Project contains 1 or 2 menus, including a title screen, but little else.	Project contains necessary menus including title, credits, game over, and settings.	Project includes menus in previous categories, as well as in game options, pause features, and more.
	Menu Styles	The project menus are not customized and use default Godot elements.	Some parts of the menus are customized, but many still contain default Godot options.	Most menu elements use custom images, text, fonts, and more to give menus an original design.	Menus form a cohesive user experience, connecting each menu together in a seamless transition of style.
	In Game UI	No in game UI is present.	Basic HUD is present to provide player information, but UI is not very dynamic or interactable.	UI provides detailed information and updates dynamically to match the gameplay.	UI is customizable by the player and follows design principles to create an immersive user experience.

Coding	**Naming Conventions**	Variables, scripts, and methods are named improperly and do not follow standard conventions.	Most names are not capitalized according to standard conventions, or names are not accurate descriptors.	Most names are capitalized properly and accurately describe what they are representing.	All names are described in detail to prevent confusion, and capitalization follows proper standards.
	Conditional Statements	No conditionals are used to gate logic paths.	One type of conditional is used, but another conditional could be better to prevent nesting.	Conditional statements are appropriately used for their given situations.	Advanced conditional structures are used to avoid nested statements, and all statements are a best fit for the situation.
	Methods	No custom functions are used in the project; only the built in Godot methods.	Godot methods are utilized to successfully implement more advanced features but lack any custom functions.	A combination of Godot methods and custom functions are used to provide unique functionality to the project.	Multiple advanced Godot methods and custom functions are used and are implemented to allow for expansion.
	Object Oriented Principles	Very few scripts are used; scripts are often responsible for doing much more than they need to.	Simple scripts are used, but often are responsible for too many items and should be split into multiple.	Script responsibilities are divided into multiple smaller scripts with single responsibilities.	Scripts and objects use advanced methods, such as abstraction and inheritance, and leave code open to extension in the future.

Ninja Planning Documents

Complete the **Release Candidate Phase – Self Assessment** portion of your Gold Belt Ninja Planning Document. Make adjustments to your game to get at least "Knight" level in each category.

Pause for **Sensei Stop #2**!

Check in with a Code Sensei. Make sure your project meets the minimum expectations. Have your Code Sensei assist in submitting your project for review.