

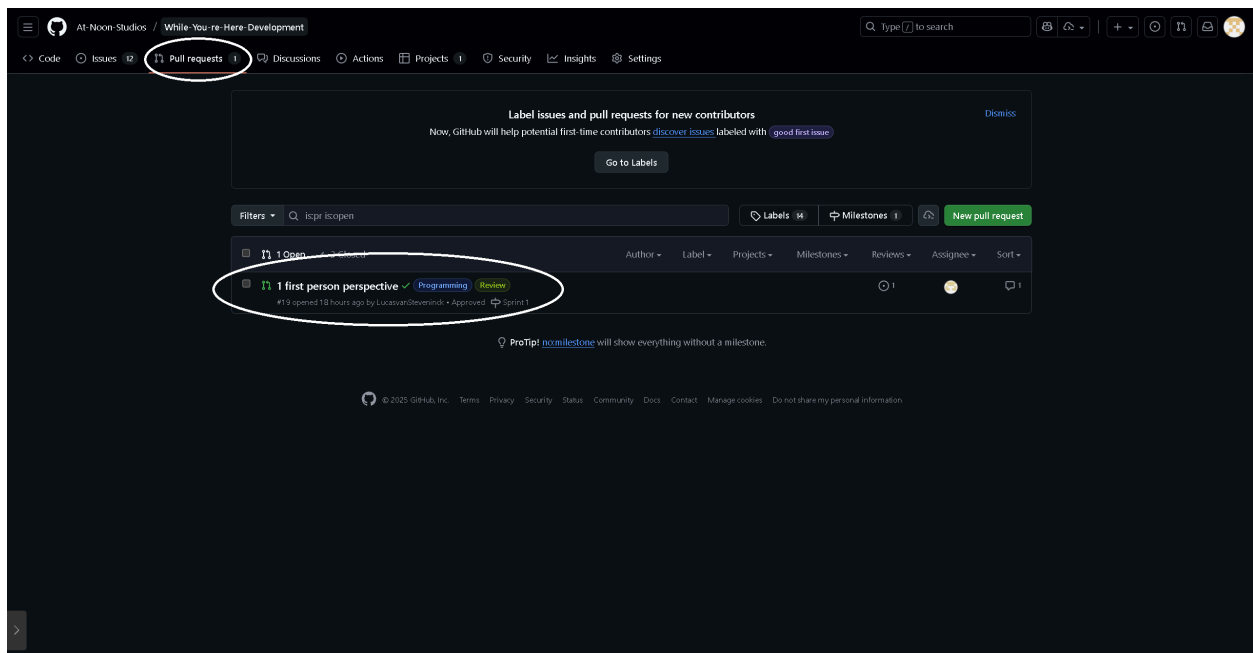


# Pull Requests

As an Integrator, your job is to review your team's work. This is done by reviewing their pull requests on GitHub. For those who are not familiar with this process, this document will guide you step by step. If you have any further questions, please contact the Integration Lead.

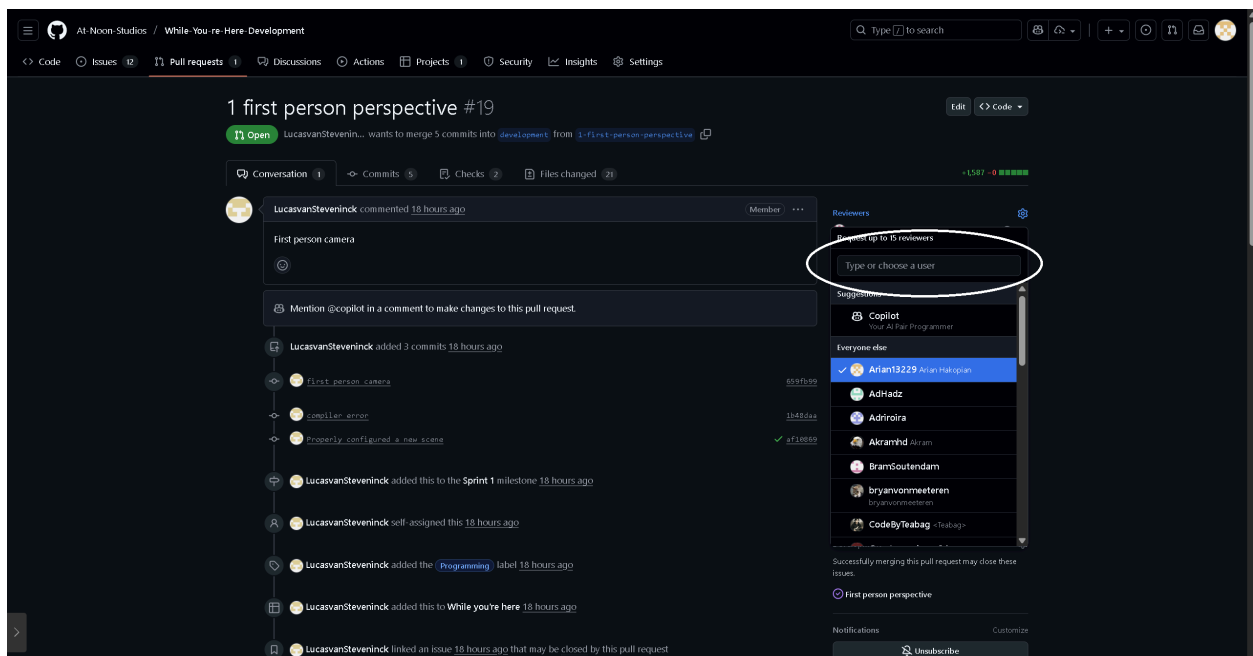
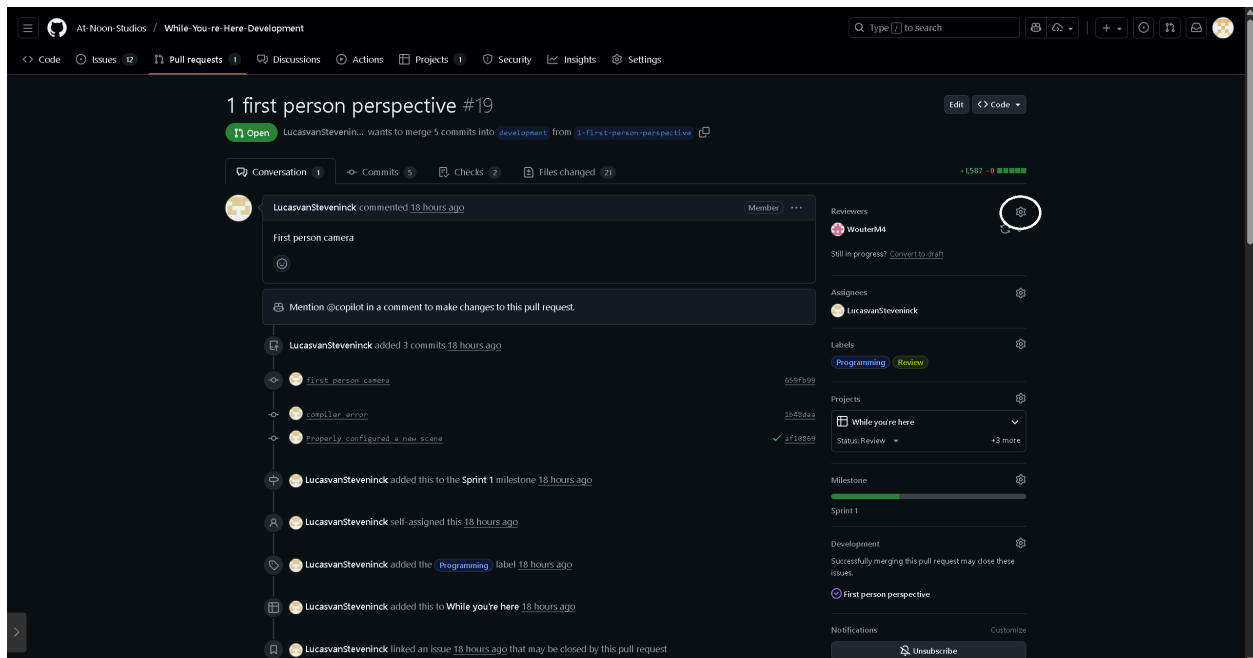
## Step 1

First, go to the "Pull Requests" tab in GitHub to view all open pull requests. Discuss with your teammates who have completed their work and created a pull request to find out what theirs is called, since there might be many open ones, making it quite chaotic to look through each individually. Once you've found the pull request you need to review, click on it. This will redirect you to the detail page of that pull request.



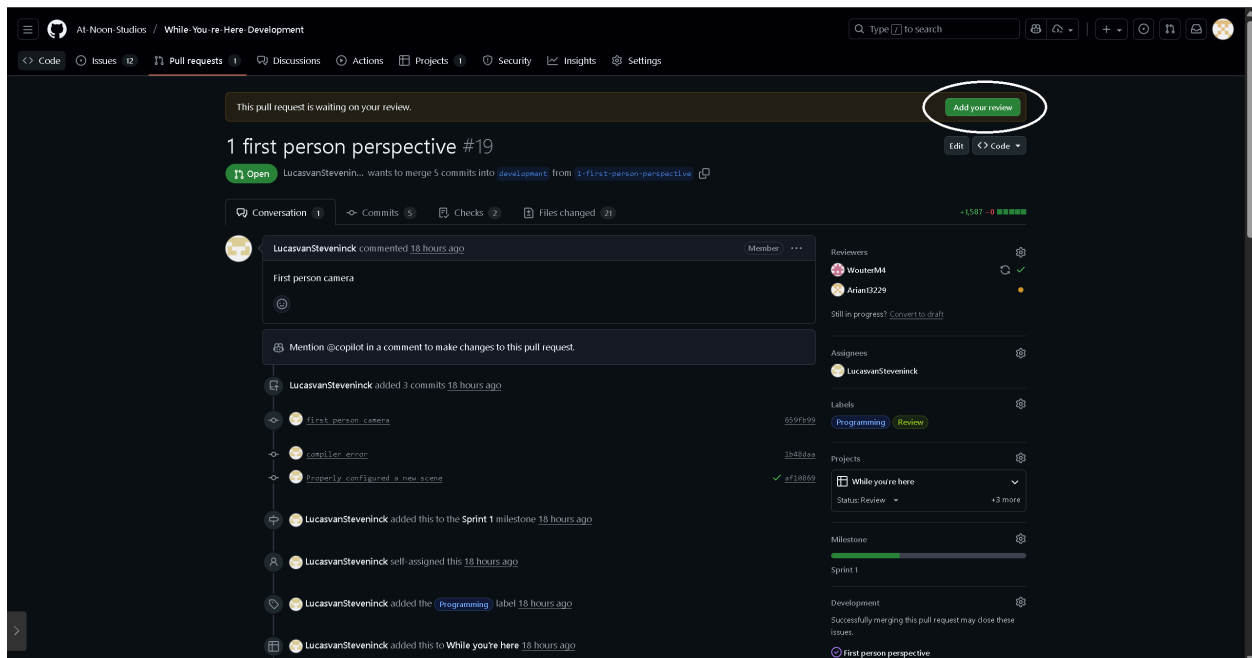
## Step 2

When opening a pull request that you want to review, the first step is to assign yourself as the reviewer. Also, make sure that the person who created the pull request is correctly assigned as the assignee. This is important for administrative reasons. To assign yourself, click the gear icon, search for your GitHub username, and then select it.



## Step 3

After assigning yourself, you should see a message at the top of the page indicating that the pull request is waiting for a review. Click the button that appears with this message to add your review. If the message doesn't appear at the top of the page, try reloading the page. It should show up after that.



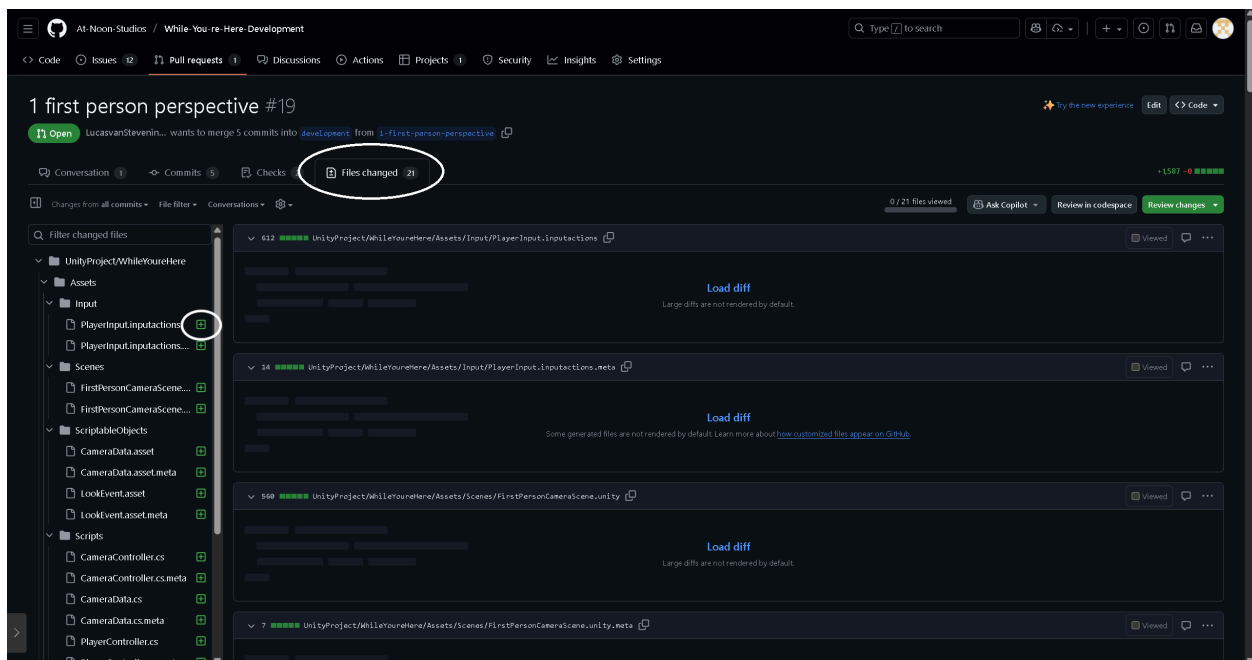
## Step 4

After you click the button to add your review, GitHub will take you to the Changed Files tab. In this tab, you can look through each file that has been modified. You can also see whether files have been added or deleted. Files that have been added have a "+" icon next to the file name, and files that have been deleted have a "-" icon next to the file name.

Your task at this stage is to go through each file and check what has been added or removed, and whether these changes match the person's assigned task. If something was deleted that shouldn't have been, the pull request should not be accepted.

Each team's pull requests will look different. For programmers or integrators, the focus is mostly on reviewing code. For artists, you should review the art. If the art

does not appear in the Changed Files tab on GitHub, open the file in the branch itself and review it there.

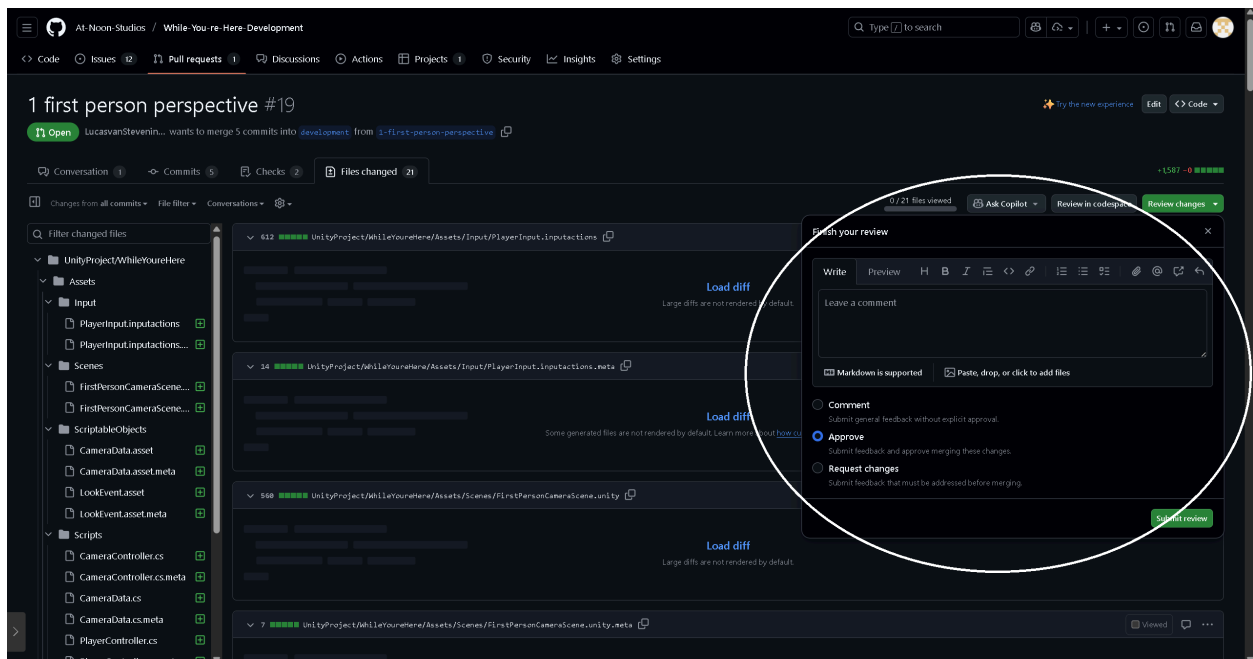
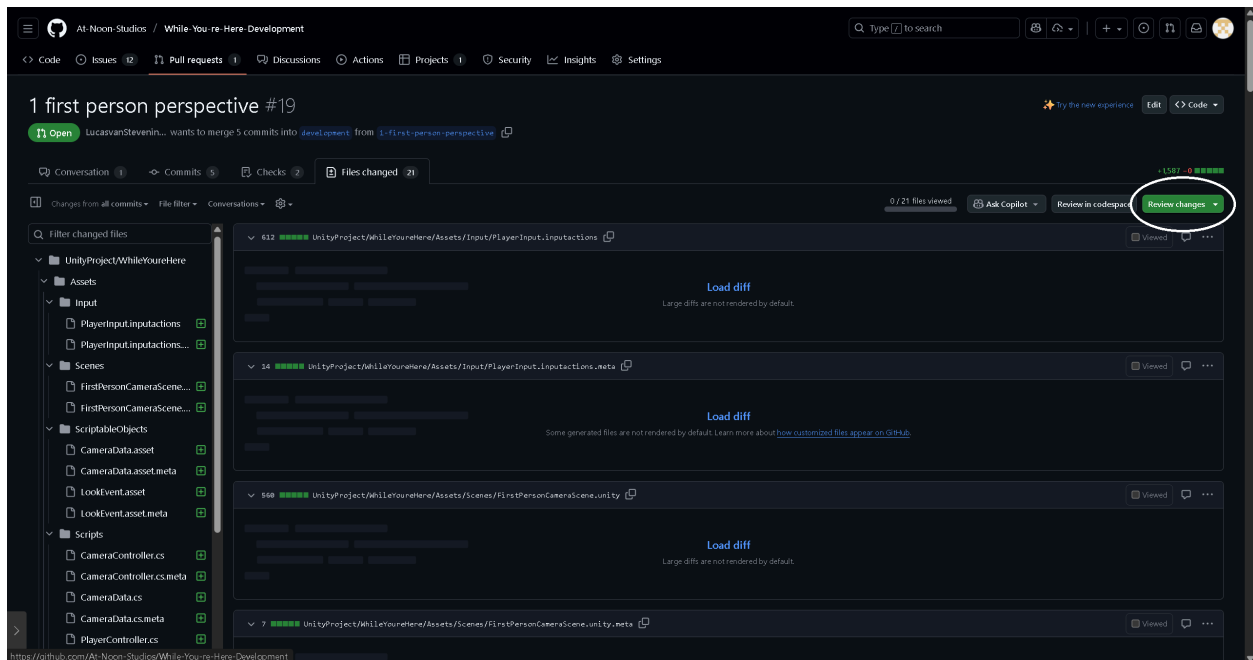


## Step 5

After finishing your review, click the “Review changes” button. This opens a dropdown menu where you can choose between three options.

- **Comment:** Select this option if you want to leave general feedback without requesting any specific changes.
- **Request changes:** Select this option if something in the pull request does not match the task requirements and needs to be fixed before approval.
- **Approve:** Select this option if the pull request is exactly as it should be and no further changes are needed.

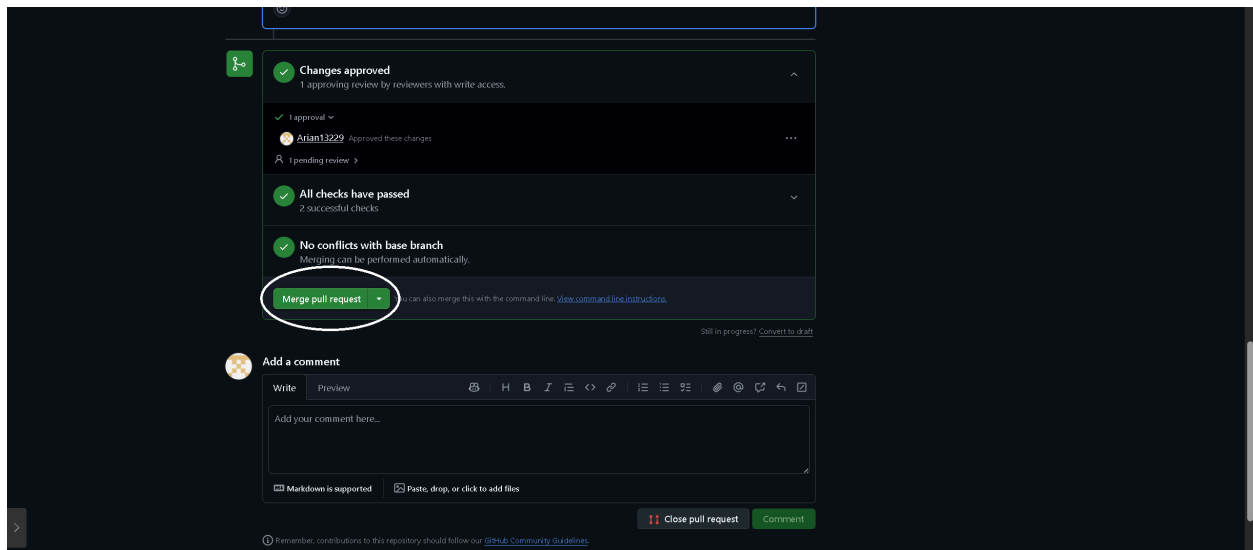
Note: You should always leave a comment when selecting Comment or Request changes. If you would like to leave a comment when selecting Approve, you can do that as well.



## Step 7

When choosing to approve the pull request, you will be taken back to the Conversation tab of the pull request. Here, you have the option to merge the pull request or leave it unmerged.

As an Integrator, you should merge the pull request because it has been reviewed and approved, meaning the changes meet the project's requirements and are ready to be added to the dev branch. Merging finalizes the process and integrates the contributor's work into the project.



## Step 8

After merging the branch, you will get the option to delete it. Once you delete the branch, you can still recover it if needed. You should delete the branch after merging because merging means the task has been successfully completed and the changes are now part of the dev branch. Deleting the branch helps keep the repository clean and avoids clutter from old branches.

**Merged** 1 first person perspective #19  
Arian13229 merged 5 commits into `development` from `1-first-person-pers...` [now](#)

Arian13229 requested a review from WouterM4 5 minutes ago

Arian13229 dismissed WouterM4's stale review 5 minutes ago

sorry dit is ff een test

Arian13229 approved these changes 1 minute ago [View reviewed changes](#)

Arian13229 left a comment [Member](#) [...](#)

Perfection

Arian13229 merged commit `ab49de6` into `development` [now](#)  
2 checks passed [View details](#) [Revert](#)

**Pull request successfully merged and closed**  
You're all set — the `1-first-person-perspective` branch can be safely deleted. [Delete branch](#)

**Add a comment**

Write [Preview](#)

Add your comment here...

Markdown is supported [Paste, drop, or click to add files](#)

[Comment](#)

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

**ProTip!** Add comments to specific lines under [Files changed](#).