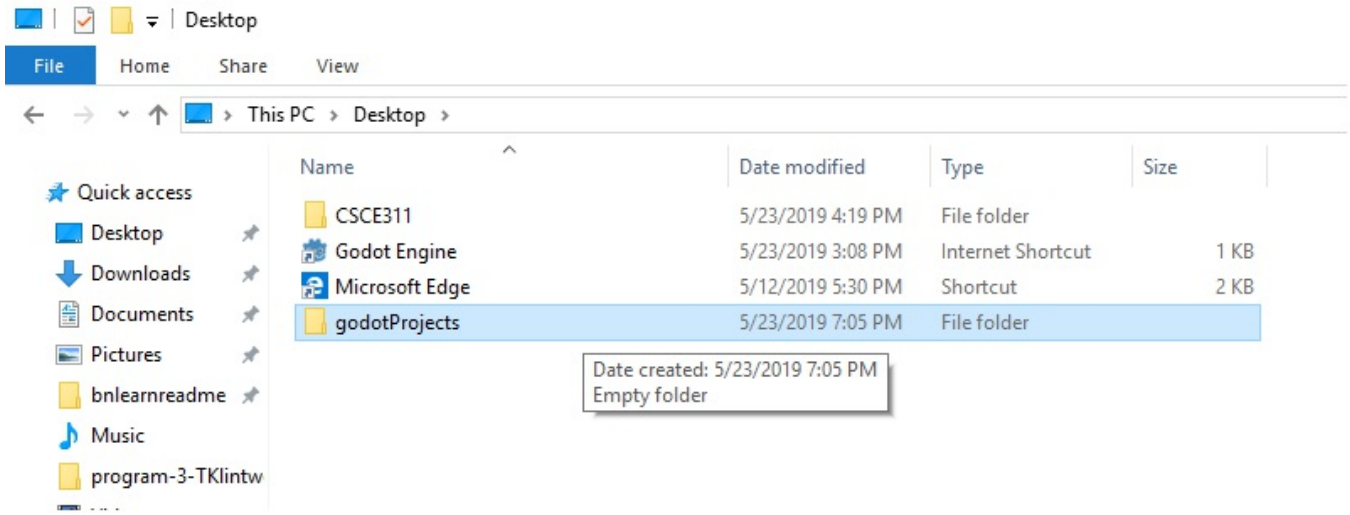
Basic Github Workflow

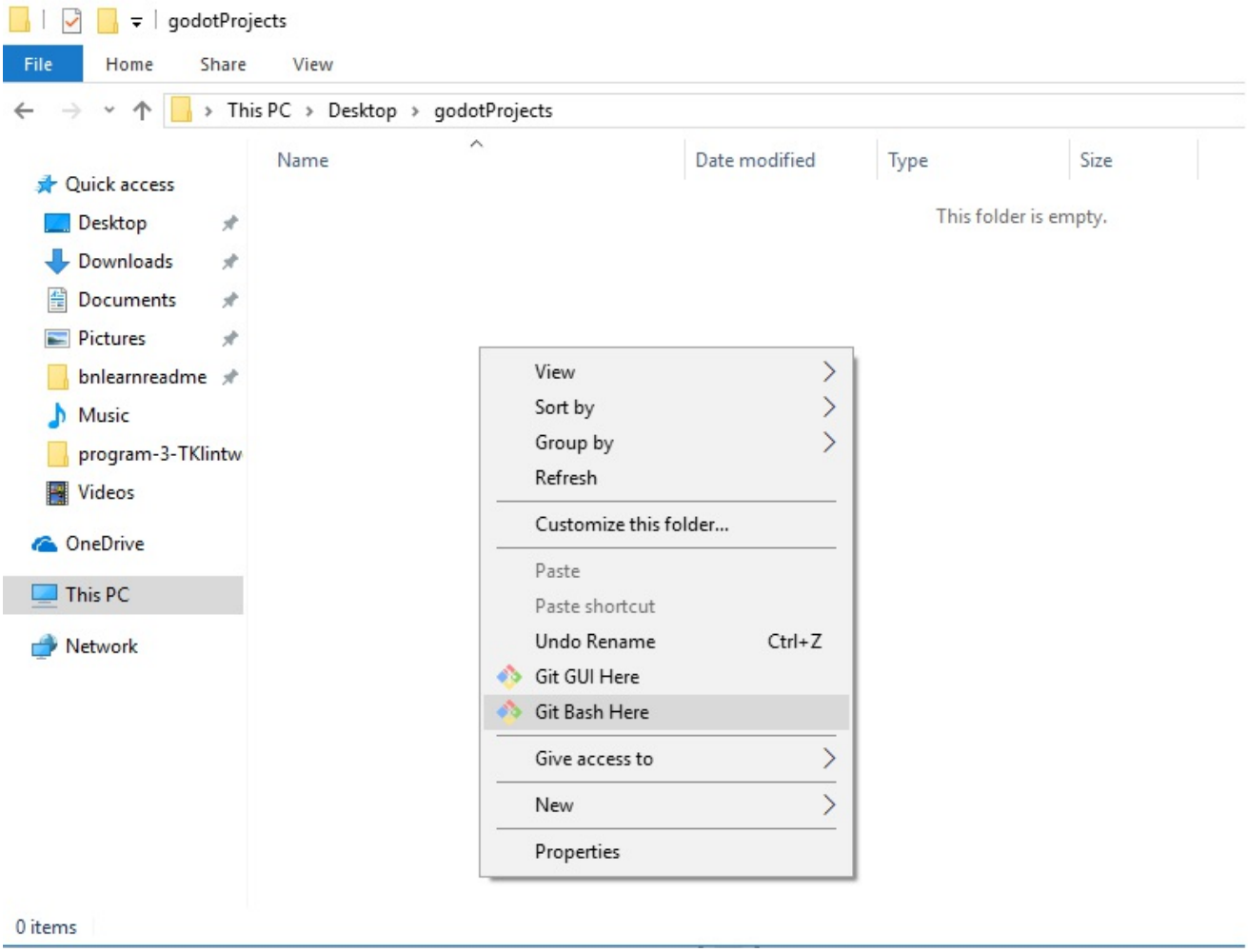
Setting up Git and cloning the repository

Go to https://git-scm.com/downloads and choose the appropriate version.

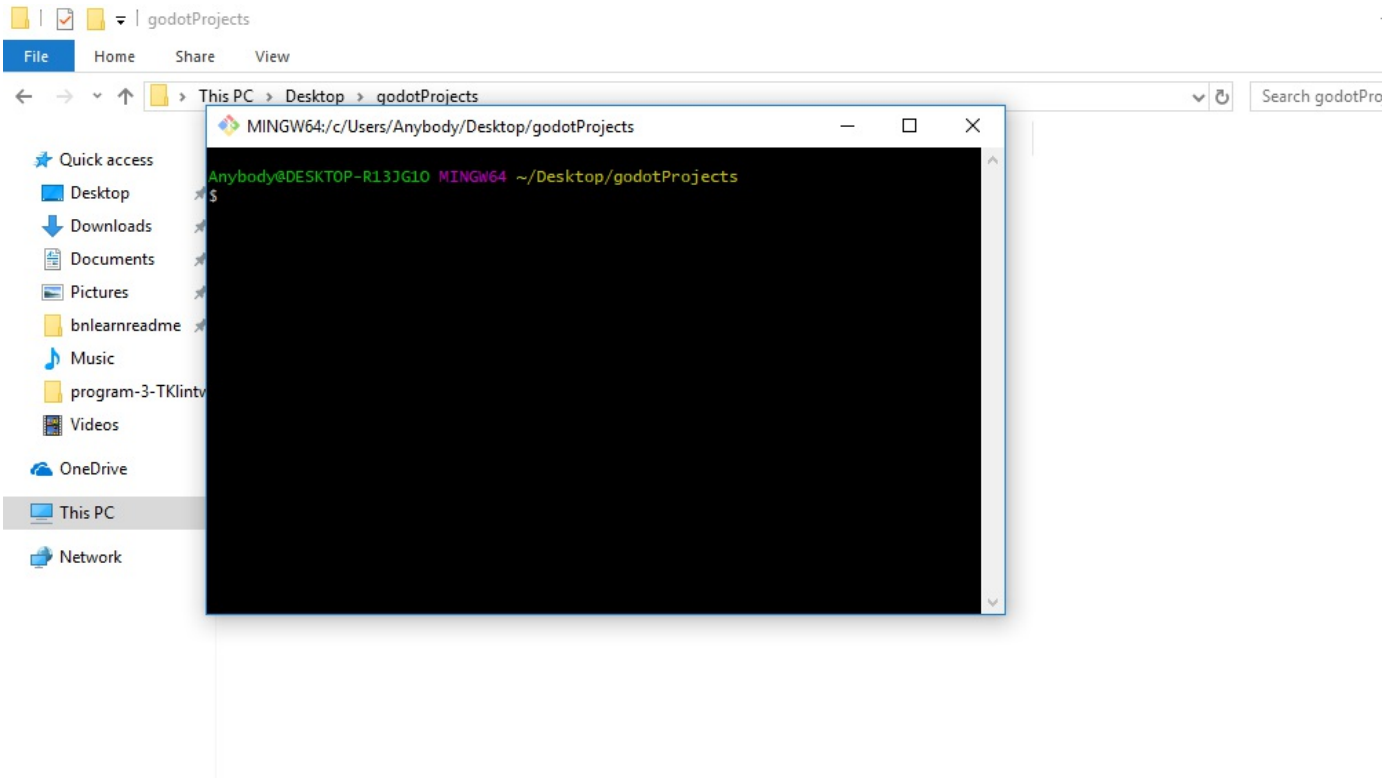
Once you have git installed, navigate to where you'd like the project folder to be (perhaps in a folder called "godotProjects" on your Desktop).



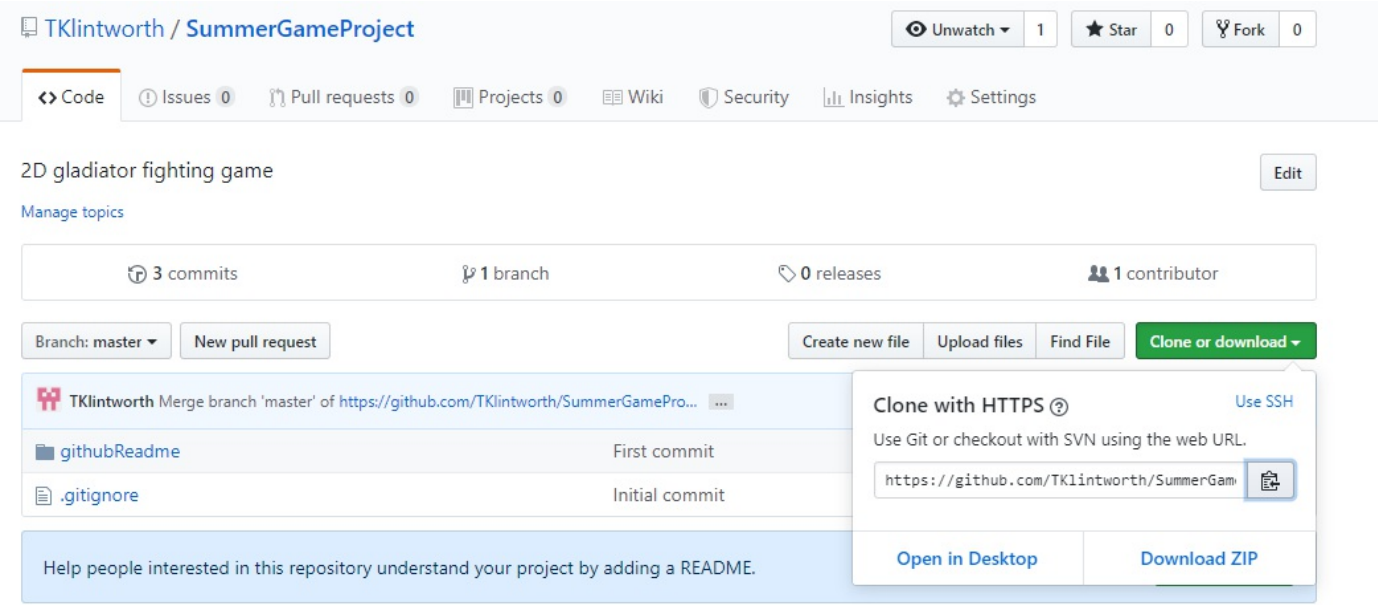
Then open that folder, in my case "godotProjects", then right click inside anywhere. You should see an option to open "Git Bash Here".



You should see a terminal like this:



Go the the github page for the project and click the button to the top right that says **“Clone or download”** as shown in the picture below. Click the button to the right to copy the link, or copy the link normally.



With the link copied, return to the command line. Type:

```
git clone [insert copied link here, without the brackets]
```

Now you should have the project downloaded.

The routine

- Create a local branch
- Make changes locally (made a new sprite, edited a file)
- Add the changes to your branch
- Commit the changes to your branch

- Push the changes to the remote version of your branch (the one on the website)
- Maybe open a pull request to add your changes to the master branch

Creating your local branch

In your project folder (the one youve previously cloned), right click and open the Git Bash. To create a new branch, type:

```
git checkout -b "your-branch-name"
```

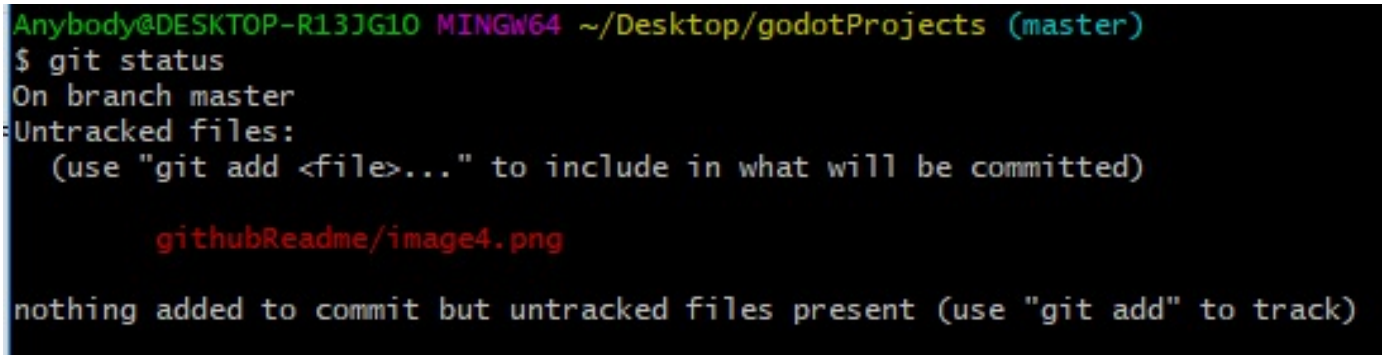
as shown here

Checking to see if you have local changes

If you've been editing files, you will need to check which files you've changed to determine which files you'd like to send to the github. To see what's changed since you've last sent, type:

```
git status
```

You should get a list of red text as shown below:



Adding your local changes

If you'd like to add every red file, type:

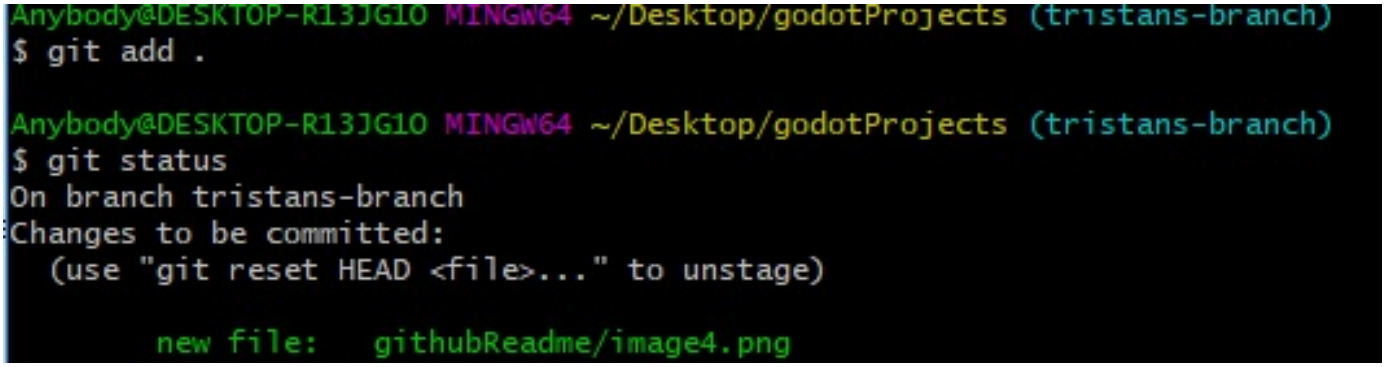
```
git add .
```

If you'd like to add only specific files, type:

```
git add your-file-name.exe
```

Individually for each file you'd like to add.

After you're satisfied, if you type ***git status*** again, you should see green text for the files you've added, as below:



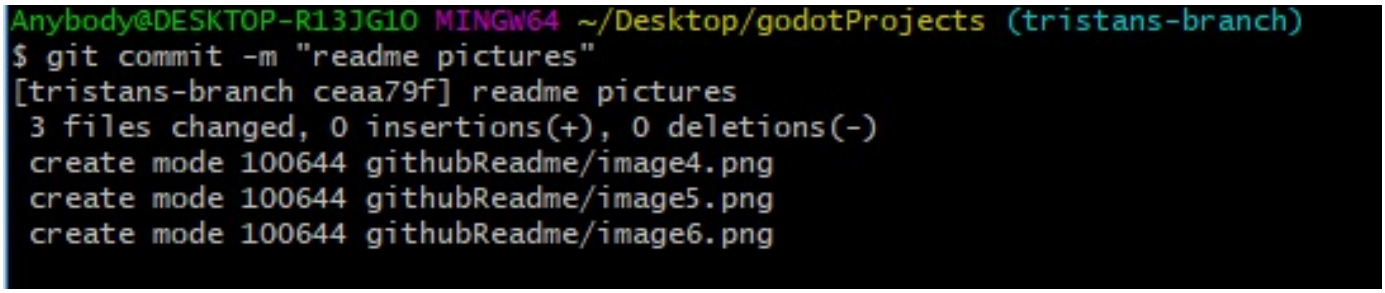
Commiting to what you've added

To actually do anything with what you've added, youll need to **commit them to your branch**, which just means putting them on your branch along with a message saying what you've added.

To commit the files you added, type:

```
git commit -m "my commit"
```

Make sure to have the quotes around "my commit" as shown.
Where "my commit" is a short message that describes what you've changed.



Pushing

Now that you've committed all the files you changed to your branch, you need to send everything you've done so far to the "remote" branch, which is your branch but on the github website.

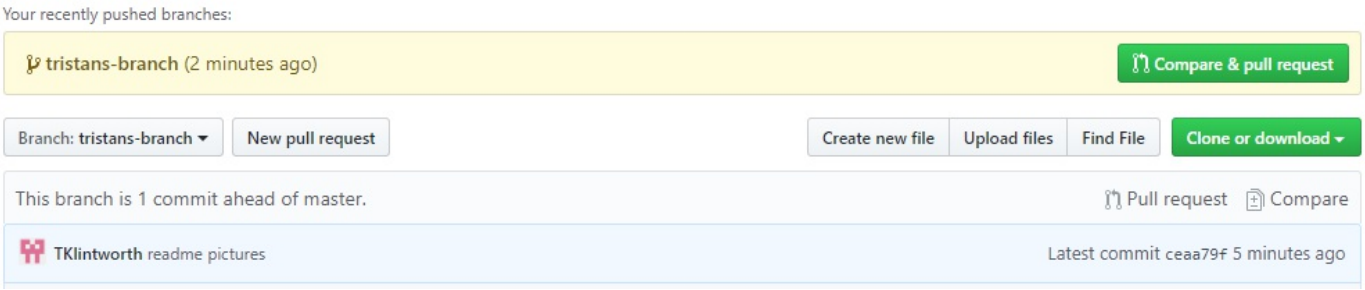
Do this by typing:

```
git push origin my-branch-name
```

```
Anybody@DESKTOP-R13JG1O MINGW64 ~/Desktop/godotProjects (tristans-branch)
$ git push origin tristans-branch
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 53.98 KiB | 3.00 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'tristans-branch' on GitHub by visiting:
remote:      https://github.com/TKlintworth/SummerGameProject/pull/new/tristans-
branch
remote:
To https://github.com/TKlintworth/SummerGameProject.git
 * [new branch]      tristans-branch -> tristans-branch
```

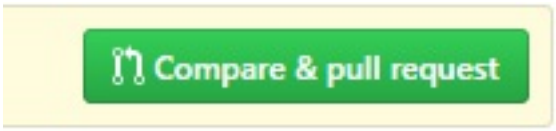
That’s it!

My example push looks like this on the website:



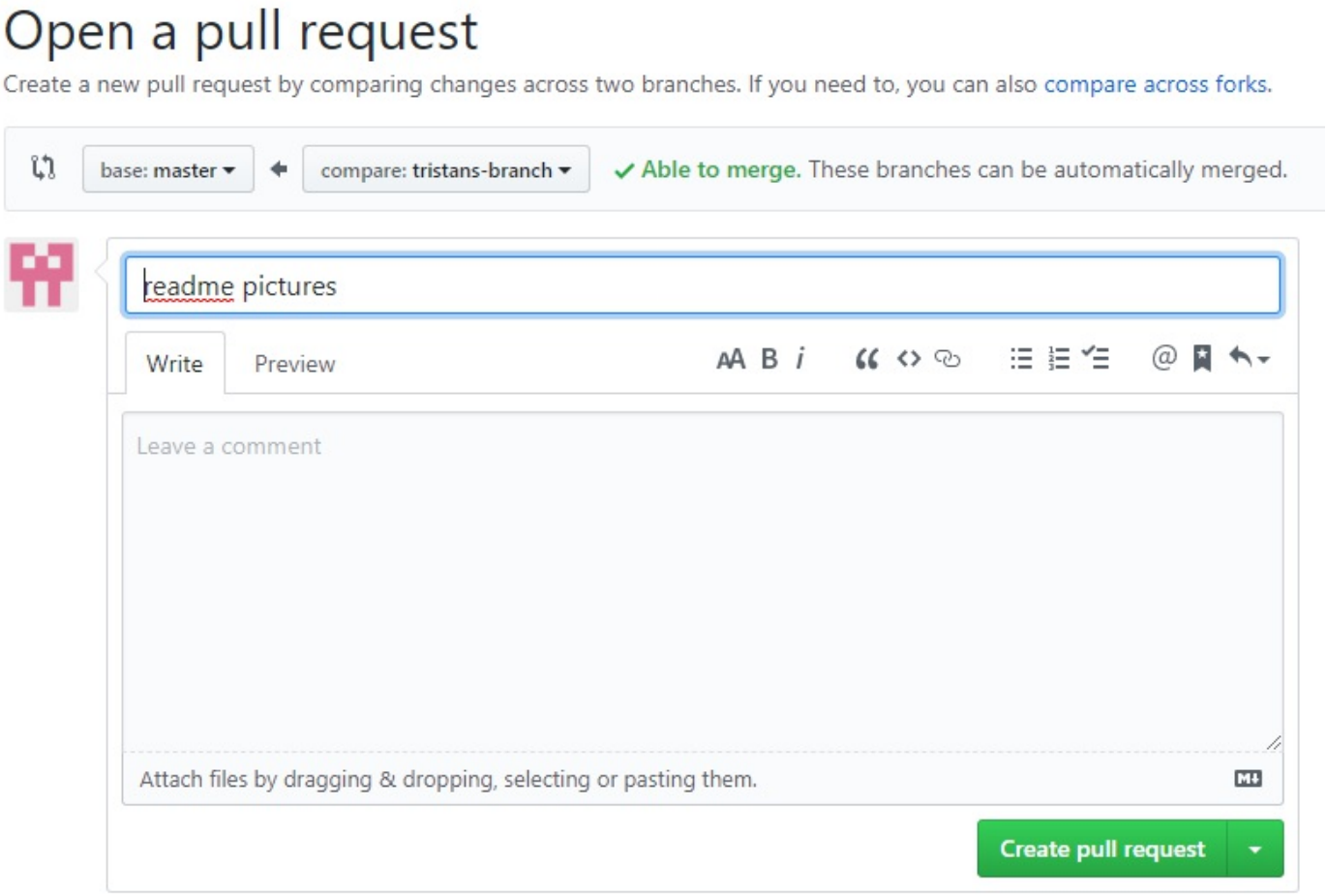
If you feel like everyone should have your changes:

Click the button shown below to create a request to pull your branch to the master



Then press the button on the bottom right where it says “Create Pull

Request”



Then wait for review and someone will merge it with the master and everyone will have your changes.

That’s it.
