# Code Conventions

## 1 Follow [C# naming convention.](https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/coding-conventions)

In summary:

Identifier names

* Almost all identifier names should use PascalCase.
* Private variables should use \_camelCase with an underscore at the start.
* Function parameters and local variables should use camelCase without the underscore.

Comments

* Comments should start with a capital letter and end with a full stop.
* Comments should appear on their own line, not tagged onto the end of a line of code.

## 2 Make sure each function is short, appropriately named and only does **one** thing.

## 3 Use OOP principles.

Encapsulation (classes should hide their data from other components)

Abstraction (classes should have interfaces to interact with other components)

Inheritance (classes should be organised into parent-child hierarchies)

Polymorphism (this is a benefit of the above, whereby children can be treated the same as their parents)

## 4 Write documentation in our shared folder for any features you’ve contributed to.

See the Player Hud and Particle Effects documents for examples.

# Continuous Integration/Continuous Deployment (CI/CD) Workflow

## 1 Checkout a new branch from main.

git checkout main

git checkout -b my-branch-name

## 2 Implement your new feature.

## 3 Test it.

Make sure it compiles without errors.

Run it to make sure you haven’t introduced any major new bugs. Make sure to do this…

* Single player within Unity; and
* Multiplayer (either with another teammate or using a built version of the game).

## 4 Stage your changes, make a commit and push to the remote repository.

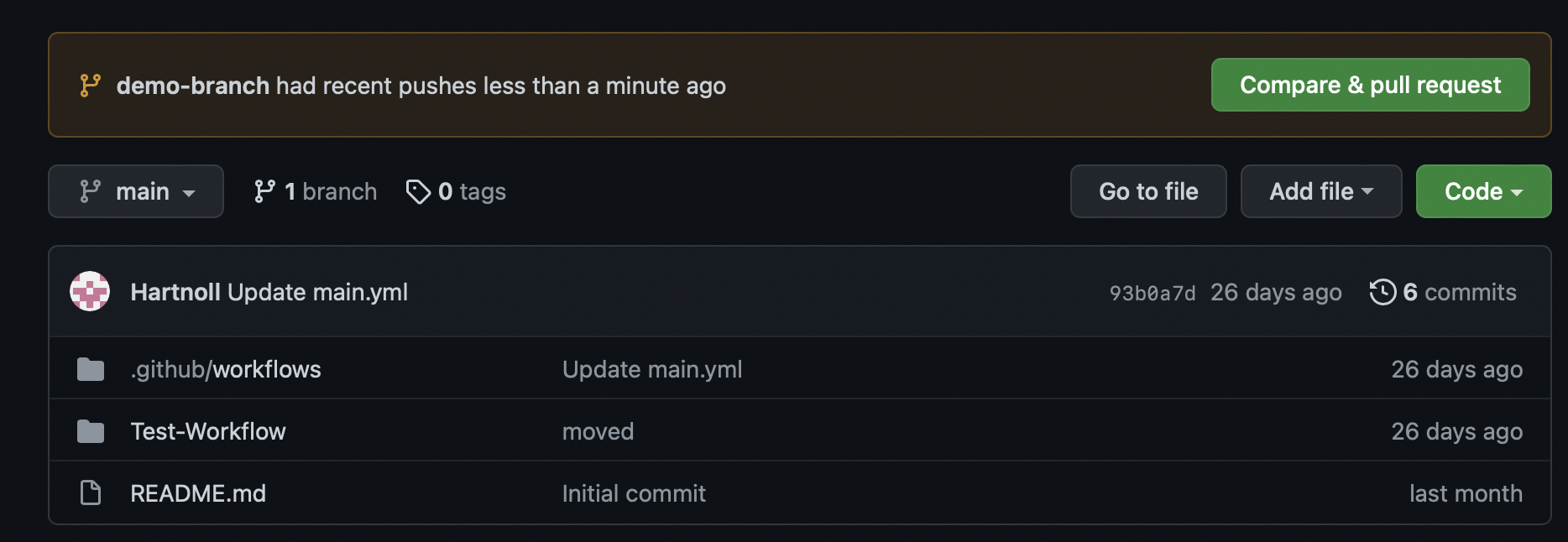
git add .

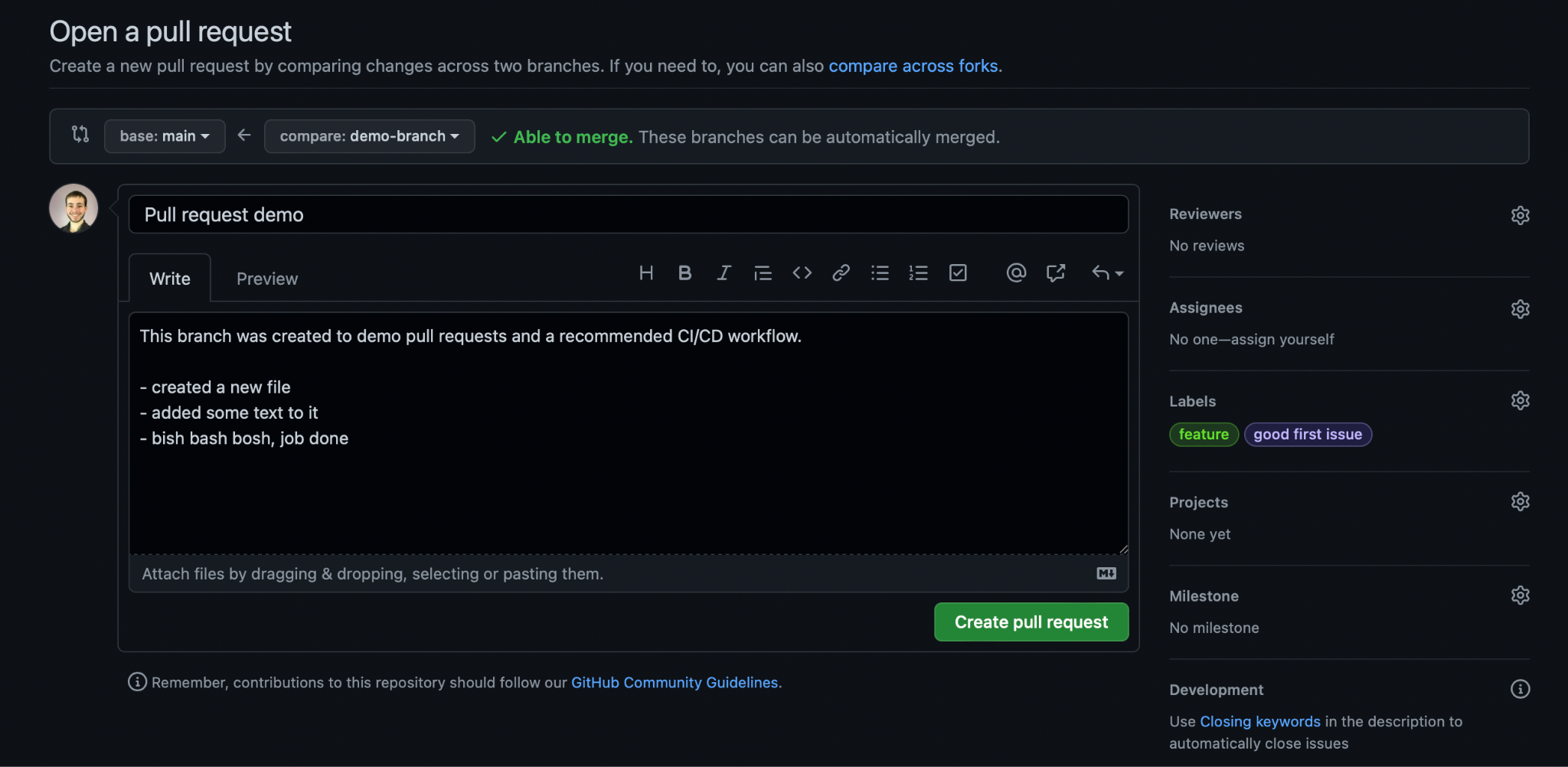
git commit -m “This is a useful message about my changes”

git pull

git push

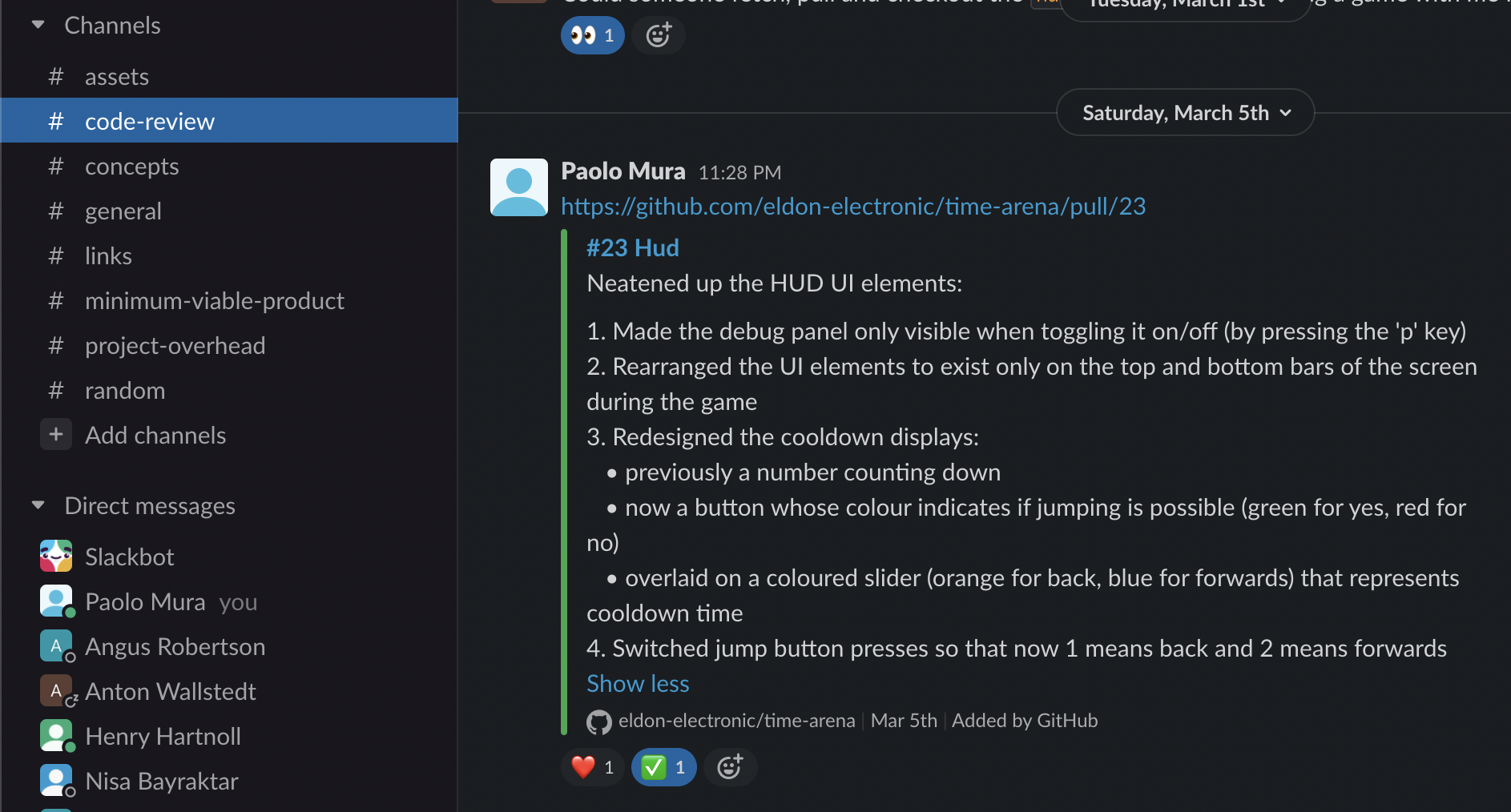
## 5 Open a pull request on GitHub.





Make sure to include a title and short description that summarises your changes.

## 6 Post a link to this pull request on the code-review slack channel.



## 7 Someone else should assign themselves as a reviewer.



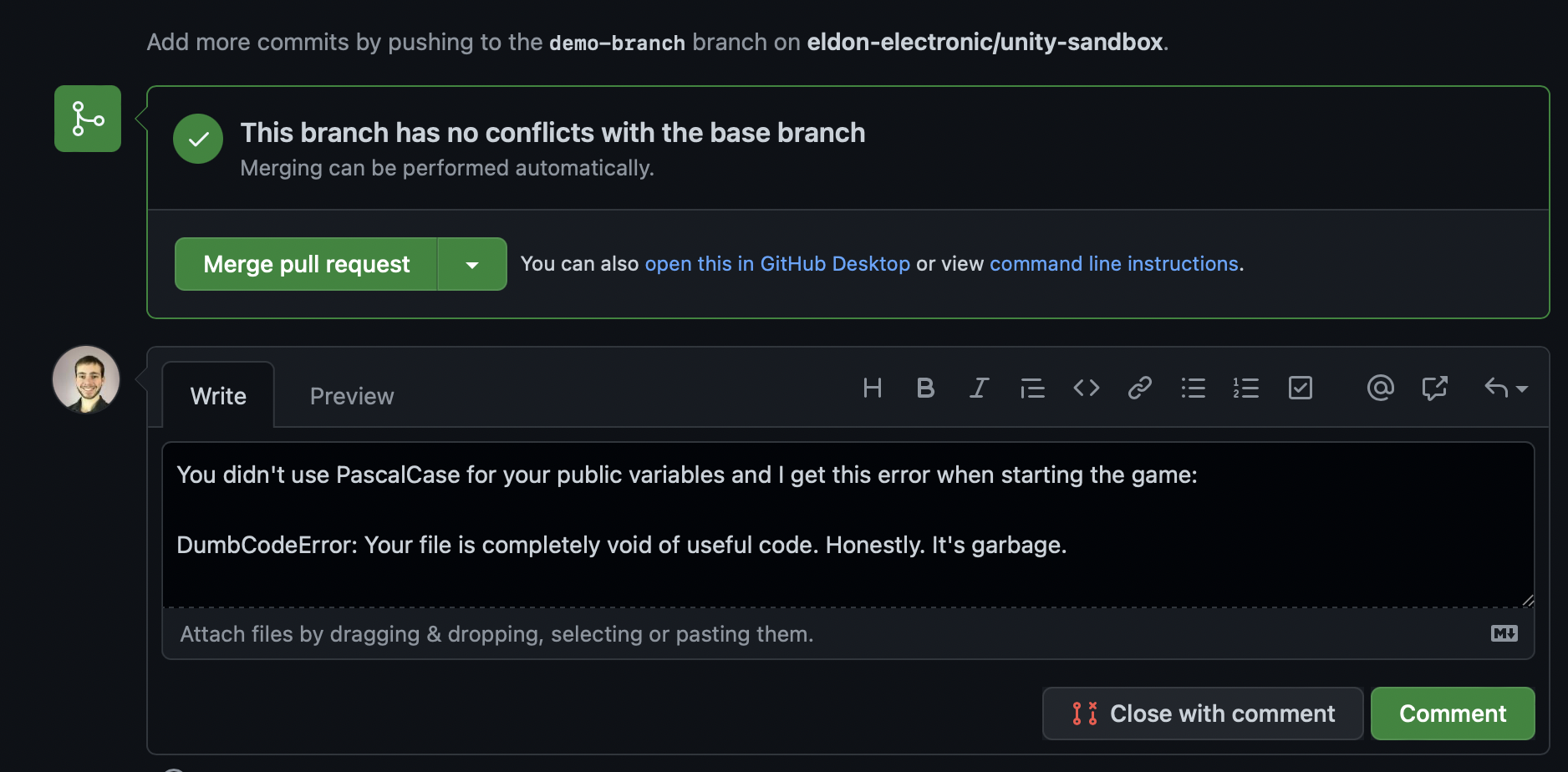
## 8 They should then do a manual code review.

Check out this branch.

Play the game to test for any obvious bugs.

Skim over the code changes to make sure they followed the code conventions above.

## 9 If satisfied, they should merge the branch into main. Otherwise they should either post a comment under the pull request thread on GitHub…



… or on the comment thread in Slack.

