B0 - HUB

B-INN-000

Git / Github

Introduction a git et github

Git / Github

- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

Fork [Github]

- fork this repository

Template [Github]

- clone the repository you have forked
- create a template that will be used when you will create an issue docs
- create a template that will be used when you will create a pull requests docs

Template test [Github]

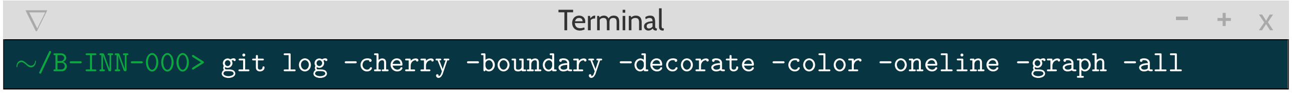
- create an issue on your repository
- create a branch named `github/test-pull-requests`
- create a pull request on your repository (branch `github/test-pull-requests` -> `main`)

Branch [Git]

- create a branch named `dev`
- create a branch named `feat/new-feature-a` that starts from `dev`
- create a file named `CONTRIBUTING.md` and `git add`+`git commit` this file
- create a branch named `feat/new-feature-b` that starts from `dev`
- create a file named `CONTRIBUTING.md` and `git add`+`git commit` this file
- add some random characters to `CONTRIBUTING.md` and `git add`+`git commit` this file

- create a branch named `feat/new-feature-a-b` that starts from `feat/new-feature-a` and merge `feat/new-feature-b`

for now, if you run:



```
~/B-INN-000> git log -cherry -boundary -decorate -color -oneline -graph -all
```

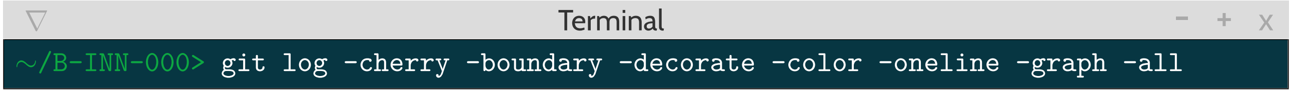
it will be something like this:
imageCenter(TODO.png, 350px, 30)

- push the branch `feat/new-feature-a-b` and create the pull requests (branch `feat/new-feature-a-b` -> `main`)

Rebase vs Merge [Git]

- read more about Rebase vs Merge: stackoverflow stackoverflow
- create a branch named `feat/new-feature-a-b-rebase` that starts from `feat/new-feature-a` and rebase `feat/new-feature-b`

re run the following command



```
~/B-INN-000> git log -cherry -boundary -decorate -color -oneline -graph -all
```

- enjoy the difference

Hooks [Git]

- read more about hooks: git-scm
- create a file named `hooks/pre-commit` and add the command to execute your norm check
- create a file named `hooks/pre-push` and add the command make to be sure that your code is ready to be pushed

GITHUB ACTIONS [GITHUB]

- read more about Github Actions: github
- create a workflow that will compile your code evey time you push on a branch
- create a workflow that will check your norm evey time you create a pull request
- create a workflow that open a issue when something goes wrong in your workflow