Open Source SW & Lab - Summer 2023
7. GitHub

Walid Abdullah Al
Computer and Electronic Systems Engineering
Hankuk University of Foreign Studies

Computer Vision Lab
Hankuk University of Foreign Studies

Based on:
Pro Git (2022) by Scott Chacon, Ben Straub

GitHub

- **Largest host for Git repositories**
- **Central point of collaboration for millions of developers**
- **Main usage in OSS development**
  - Git hosting
  - Issue tracking
  - Code review
  - Etc.

GitHub: contributing to a project

- If you want to contribute to a project
- But do not have push access
- You can "fork" the project
- **Forking projects**
  - GitHub will make a copy under your account
  - With full access
  - Thus, projects do not need to add collaborators
- **After pushing to a forked project**
  - Create pull request
  - This starts a discussion thread with code review
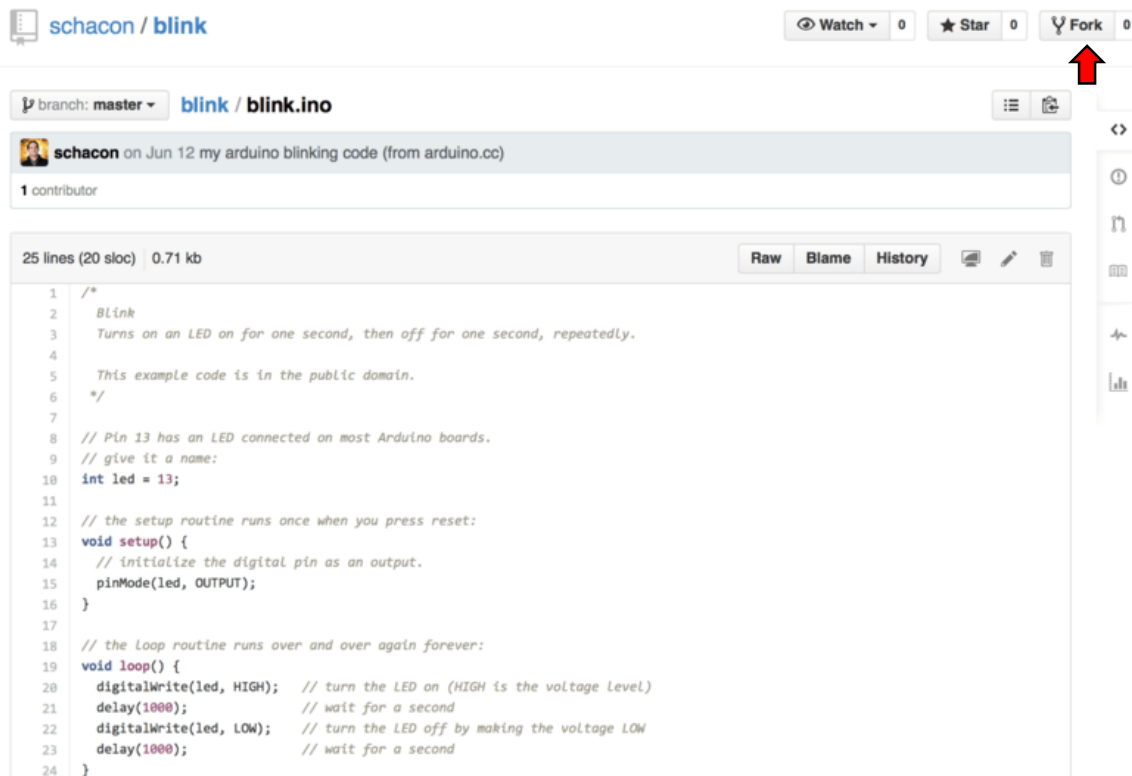 - The owner and contributors communicate
  - The owner merges it in, when he is happy

The GitHub Flow

- **Centered on**
  - Pull requests
  - Topic branches workflow
- **General flow:** integration-manager workflow
  - Fork the project
  - Create a topic branch
  - Make some commits
  - Push this branch to your fork
  - Open a Pull Request
  - Discuss, and optionally continue committing
  - The project owner merges or closes the Pull Request
  - Sync the updated master back to your fork

Example

- Tony is looking for code to run on his Arduino programmable microcontroller and has found a great program file on GitHub

Problem:
blinking too fast

Possible improvement:
Increase delay to 3s

How can we contribute to the project?
Click the "**fork**" button

Fork: Clone – edit – Push back

- **After forking**
  - A copy is created at:
    https://github.com/tonychacon/blink
- **Clone the fork, create a topic branch, make the code change, and push to the fork**

```
$ git clone https://github.com/tonychacon/blink (1)
Cloning into 'blink'...

$ cd blink
$ git checkout -b slow-blink (2)
Switched to a new branch 'slow-blink'

$ sed -i '' 's/1000/3000/' blink.ino (macOS) (3)
# If you're on a Linux system, do this instead:
# $ sed -i 's/1000/3000/' blink.ino (3)
```

```
$ git commit -a -m 'Change delay to 3 seconds' (5)
[slow-blink 5ca509d] Change delay to 3 seconds
 1 file changed, 2 insertions(+), 2 deletions(-)

$ git push origin slow-blink (6)
Username for 'https://github.com': tonychacon
Password for 'https://tonychacon@github.com':
Counting objects: 5, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 340 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To https://github.com/tonychacon/blink
 * [new branch]      slow-blink -> slow-blink
```

Pull Request

- **After pushing, if go back to fork at GitHub**
  - GitHub will present the pull request option
  - You can alternatively go to the "Branches" page at **https://github.com/<user>/<project>/branches** to locate your branch and open a new Pull Request from there
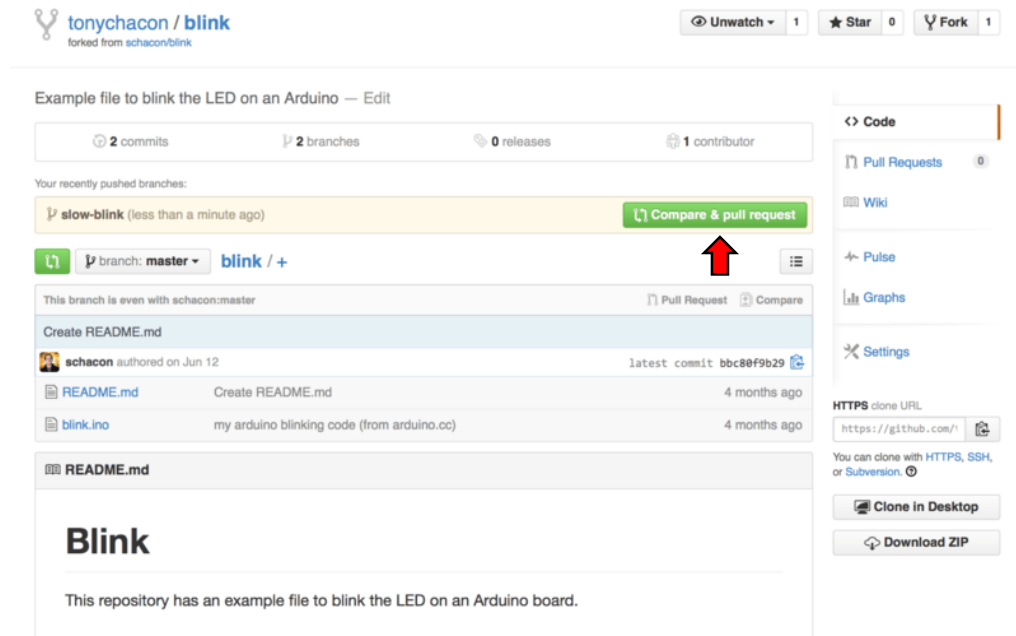
After Clicking Pull Request

- **Fill out**
  - the pull request title and description
- **Check all the commits and differences**
- **Click "Create pull request"**

Iterating on a Pull Request

- **The owner now can**
  - Merge it / reject it / comment on it

Pull Request discussion page

Do the changes and re-commit

- **Only, push to the fork**
- **Pull-request would be updated automatically**
- **However, no notification to owner**
  - Therefore, comment explicitly

Comment history

Today's team assignment

- **Maintainer/member-1:**
  - Creates a public repository in Github
  - Add pong.py file to the repository
    - (uploaded on Eclass)
  - Does not add collaborator
- **Other members:**
  - Fork the repository
  - Create topic branches to do their work
  - Create a pull request
- **Maintainer:**
  - Merge pull-requests

Task-distribution

Member1:
Gameloop
Simulate.update_obj()

Member2:
Pong.update_ball_pos()
Pong.process_fall()

Member3:
Pong.process_border_hit()
Pong.process_player_hit()

Each member has two tasks.
Flow:
- Choose a task.
- Create a branch for it.
- Commit your work.
- Create a pull request.

- Then, choose the other task and do the same

Submit on Eclass

- Screenshot of the final game-window
- Your repository link