Using Git and GitHub

Tecnología de Videojuegos

Objectives

1. Understand the need of SCM
2. Implement software development workflows with Git and Github

Bibliography

1. GitHub Guides. (Link)

Table of Contents

Version control

Version control systems

Version control systems (VCS) keep track of changes to source code. Allows multiple people to edit a project in a predictable manner.

Main open source VCS

- 1982 RCS
- 1990 CVS
- 2000 Subversion
- 2005 Git/Mercurial

There are many proprietary ones but `Git` is now the most popular one by far.
All software should be under a version control system, if not, it ain't software!

Git

What is Git?

Git is an open source distributed version control system, created by Linus Torvald.

`https://git-scm.com/`

(Interactive tutorial)

Git

Git sites

It is easier to start with free hosting sites instead of maintaining your own server.

- GitHub: public repositories (as many as you want), but private ones are not free (except for academia). It is now part of Microsoft
- Bitbucket: allow us to keep private repositories limiting the number of collaborators.
- GitLab: both public and private without limitations. It is becoming more popular.
- Others ...

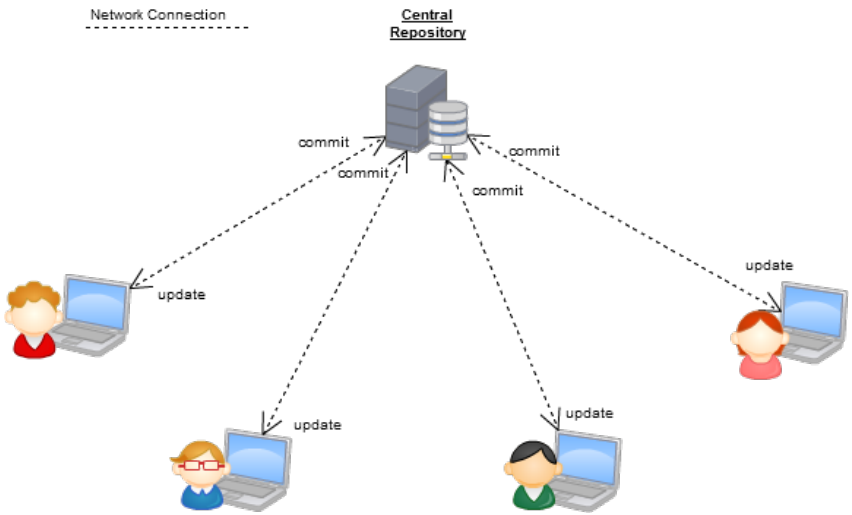
It is typically used as central repository:

- from which everyone pulls other people's changes
- to which everyone pushes changes they have made

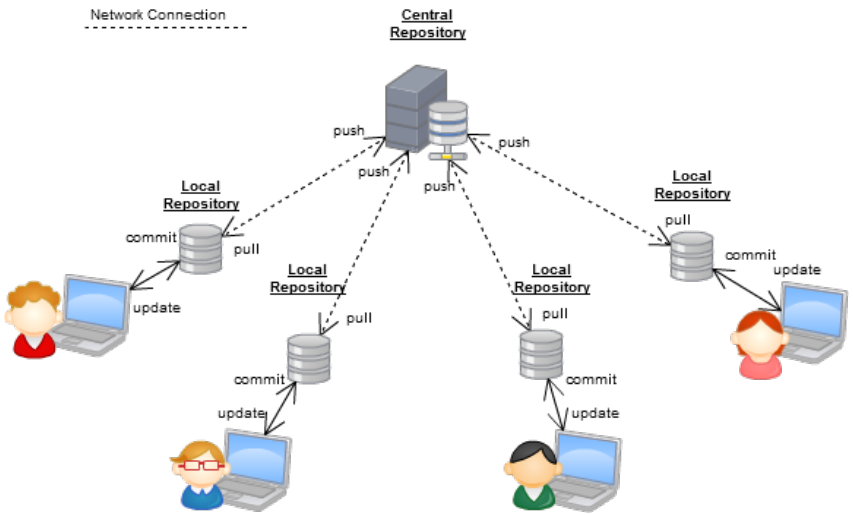
Git

Git vs. SVN (I)

Centralized (SVN)



Distributed (Git)



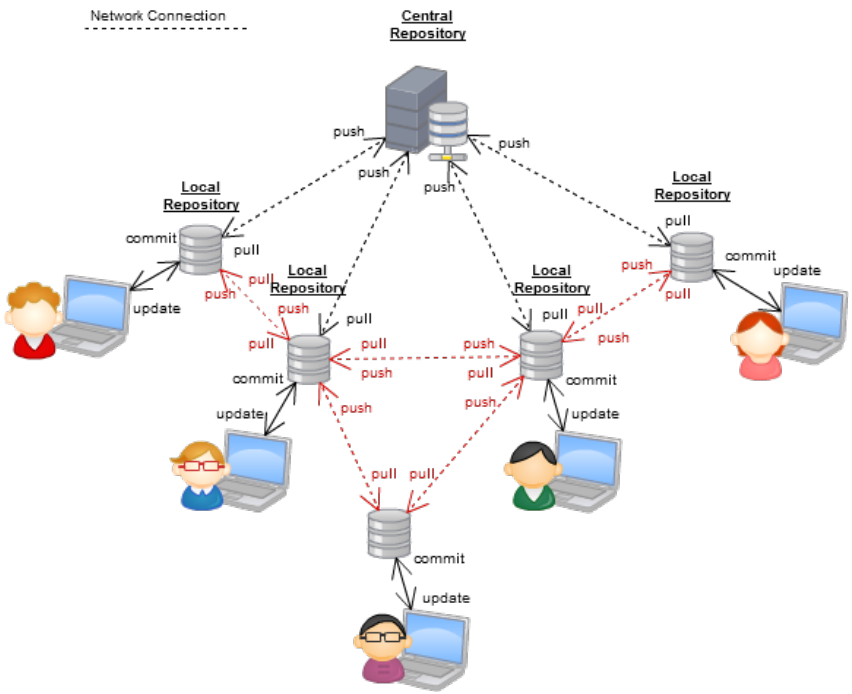
(Source)

Disclaimer: Do not pay attention to the labels of these diagrams

Git

Git vs. SVN (II)

Fully distributed (Git)



(Source)

Key Git concepts to know

- `repository` (local or not)
- `clone`
- `commit`, `push`
- `pull`, `fetch`
- `remote`, `origin`
- `merge`

Using Git

Conflicts

If two people both modify the same file, the first to push *wins*. The second person will have to pull and merge before pushing.

- Changes in different parts of a file are automatically merged
- Changes in the same part of a file cause conflicts
  - Select either your changes or remote, or a mix of the two

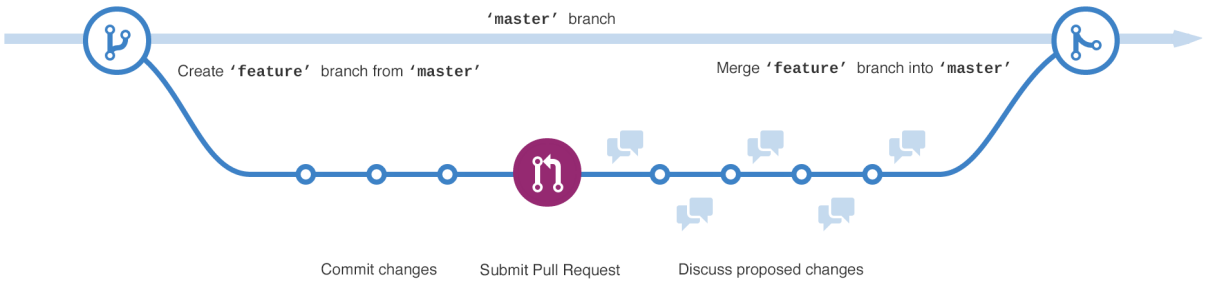
Using Git

Commits

- Every commit has an ID (its hash)

Using Git

Branches



Branches are used extensively (e.g. some like feature branches).

- A repository (local and remote) can have explicit branches
- The default branch is called master
- A merge is a fusion between two branches

Do no use branches in the project!

Using Git

Tags

A tag is a pointer to a specific point in the repository history

- Tags usually have names (e.g. “v1.1”)
- Widely used to keep and publish software releases

You might try to tag your project

Using Git

Good practices

Learn on the job: the best way to learn it is by using it.

Best practices

- Regularly push and pull (at least daily, in general)
- Don't push half-baked changes
- Don't pull if you're in the middle of a task
- Never commit temporal/intermediate files
- The master must be a clean and functional version of the project

GitHub

Features

Free Git hosting provider

- Free public repositories

User interface to Git

- Repository browser

Added value Git operations

- Gist
- Pull requests
- Collaborative tools
- Issue tracking
- Web hosting
- Integrated Jekyll processor
- Markdown integration
- Organizations

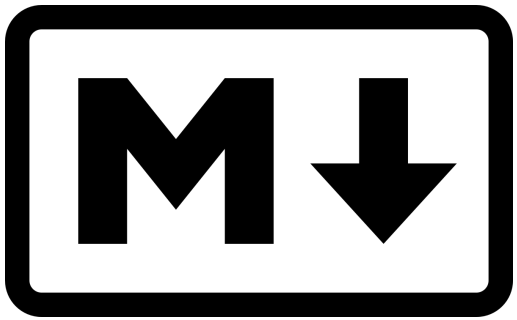
GitHub

README

Special file: README.md

- Contains information about the project
- Automatically visualized
- md means Markdown

Markdown (I)



Markdown: Trivial markup

- Simple
- Very simple
- Extremely simple
- Did I say it's simple?

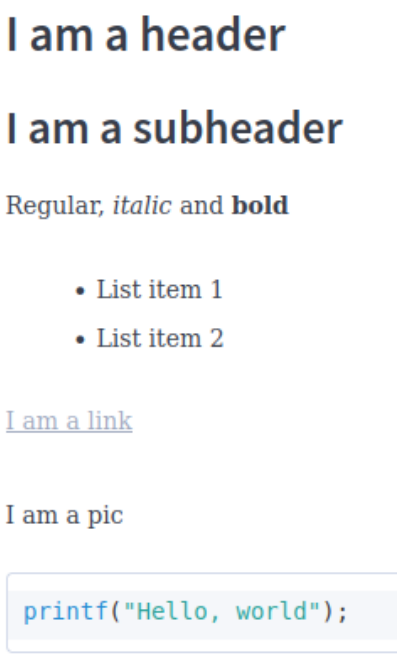
VERY powerful

- Several outputs
- Professional quality
- ... and simple!

Markdown (II)

Markdown example

```
# I am a header
## I am a subheader

Regular, *italic* and **bold**

- List item 1
- List item 2

[I am a link](http://foo.com)

![I am a pic](markdown.png)

~~~C
printf("Hello, world");
~~~
```



GitHub

GitHub Pages

Pages integrate web site in the GitHub workflow

- Creation of full web sites
- Project web site
- Documentation
- Based on Markdown (and something named *Jekyll*

GitHub locates the content to publish in three places:

- A branch named `gh-pages`
- master itself
- A folder `docs` in master
- **Page available on `https://<username>.github.io/<repository>`**

By default, Pages are disabled

- Enable them in settings

User Page. Site accesible in `https://<username>.github.io`

- The repository must be named `<username>.github.io`
- master branch