Engine Architecture & Design : Review Questions 1

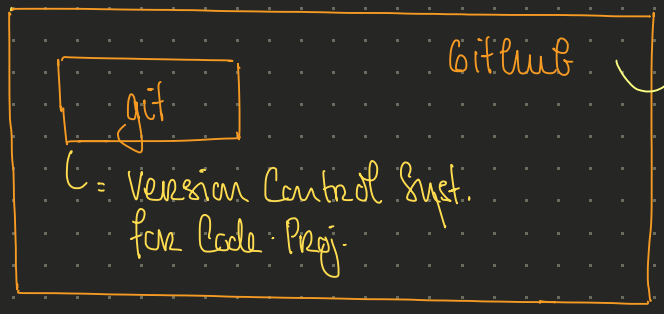
Salomé Christiani

11.11.2024

Ex 1 : Engine Setup :

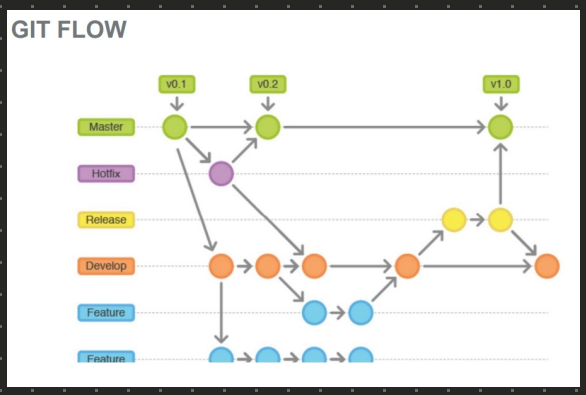
▷ What is the difference btw. Git & GitHub

git : open source github : owned by Microsoft



⇒ git flow vs github flow

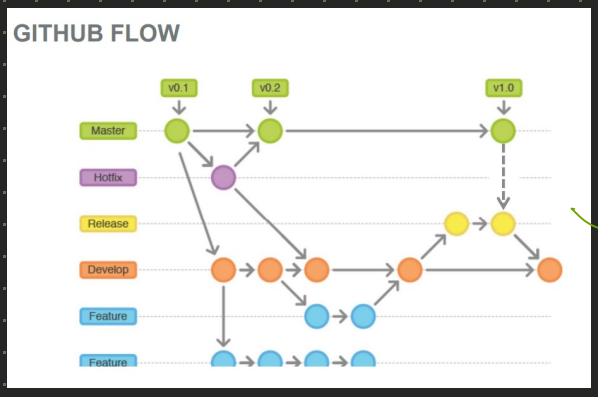
⟺ △ Difference ⟸



Git

- Version Control System
- Distributed Architecture
  - develop a project simultaneously and independently
  - Local copy of project
  - work offline and merge changes later
- Repository: can be local or remote
- Commits:
  - snapshot of the project at a specific point in time
  - store changes made to files and include a unique identifier

- Branching and Merging:
  - Flexible way to organize work
  - Naming conventions for branches: main, feature, hotfix, bugfix, …
  - Different strategies: Git Flow vs GitHub Flow
- Pull and Push:
  - Retrieve changes from a remote repository (pull)
  - Upload local changes to a remote repository (push)



constant flow of new features

master is always functional

AND

▷ Name 3 Benefits of a version control system

#1 Collaboration : local copy of the repo → merge back to a (feature) branch ← having #2 Control over changes, potential conflicts & added code, not forgetting to mention #3 multiple branches launchable for a smooth CI/CD workflow (eg. when managed with gitkraken)

▷ Name 3 alternatives to git:

SVN, Perforce, Fossil

Bitbucket (~ like Github) ← Atlassian

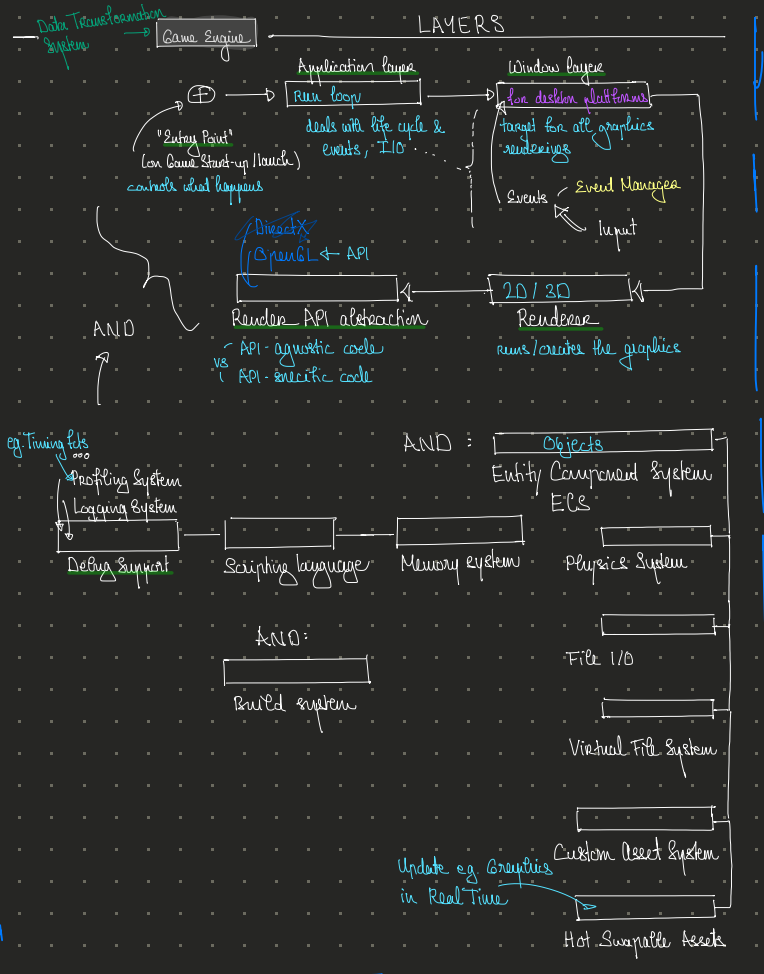
▷ Name 2 differences btw git and other version control systems

- git: distributed version control system
  - clone has its own local database of the source
- use via command-line or integrated into a large number of IDE's, git is leveraged by a number of applications (e.g. Github, Bitbucket) – therefore widely used (across industries & companies) and also well documented

▷ Name known issues related to game Engines (Unreal, Unity) and version control systems

as git clones entire databases (eg. on the download of a repo)
⇒ overwhelming for bigger projects on git
↳ large download
large database & storage demands

large in size: (example)



▷ DESCRIBE A LOGGING SYSTEM IN YOUR OWN WORDS

provides prebuilt callable functions to be used across the program for the purpose of

▷ extracting / making available program "internal" infos in nature commonly trace, warn, info, error, fatal, ...

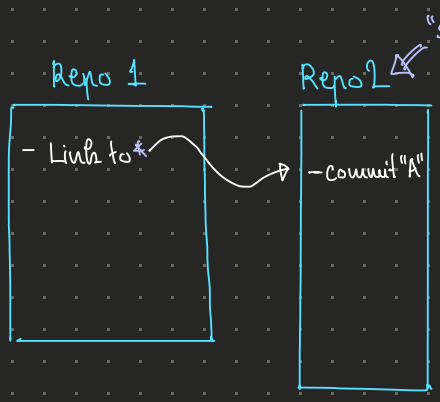
→ with presetted formatting

→ where variable and arguments can be parsed through (runtime data) as one use case)

Serves the developers of the code eg. for debugging and maintenance but also user how reference / make use of that code (included as libraries for example).

▷ WHAT IS THE DIFFERENCE BETWEEN GIT SUBMODELS AND GIT SUBTREES?

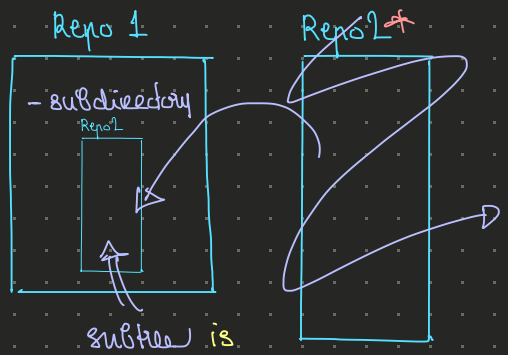
Both are methods to include one repository inside another though they are managed differently and feature a different workflow



init, update & changes (e.g. of the submodule the Repo 1 points to) all need to be performed manually

→ independent versioning

⇒ allows for more control through manually performed commands, thus higher flexibility



- embedded history for Repo2 in Repo1

⇒ not a seperate repo anymore

no seperate updating

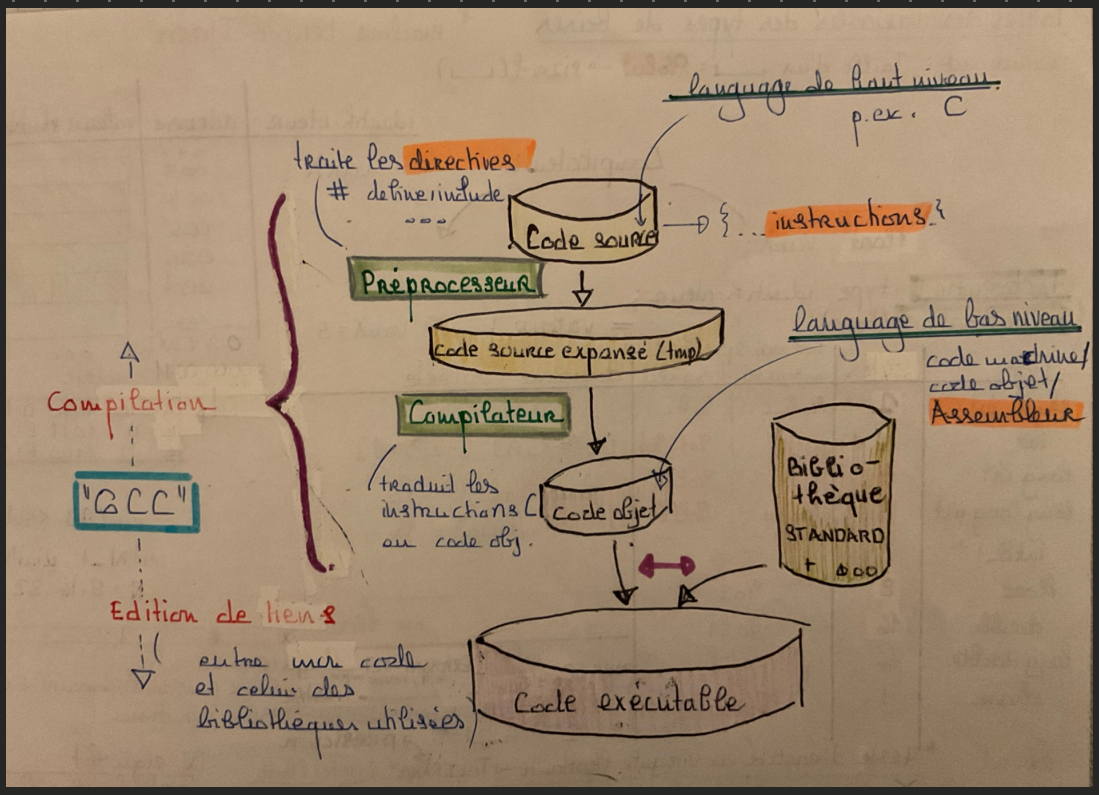
⇒ allows for less manual navigation/operation, a simplified handling of dependencies, easier maintenance (supposedly)

WHAT IS A BUILT SYSTEM IN C++?

ex. Make, CMake, Premake, Meson

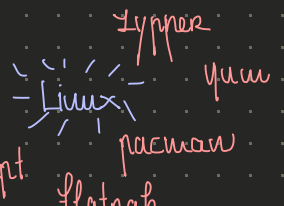
says "how individual building blocks are going to be assembled to form the application" while:

- supporting the user to keep configurations up-to-date
- recognizing & sorting out sources & their dependencies upon changes performed
- helping with cache management



(my own notes)

WHAT IS A PACKAGE MANAGER IN C++?



vcpkg (Microsoft) xrepo (OSS) Conan (JFrog)

- handles dependencies of a project for precompiled, preexisting sources used by the program / project
- handles downloads from repositories and version (in-) compatibilities / conflicts

⇒ access to download a wide range of programs, development tools & (C++) libraries from a repo