Software Studio
軟體設計與實驗

Git

Hung-Kuo Chu
Department of Computer Science
National Tsing Hua University

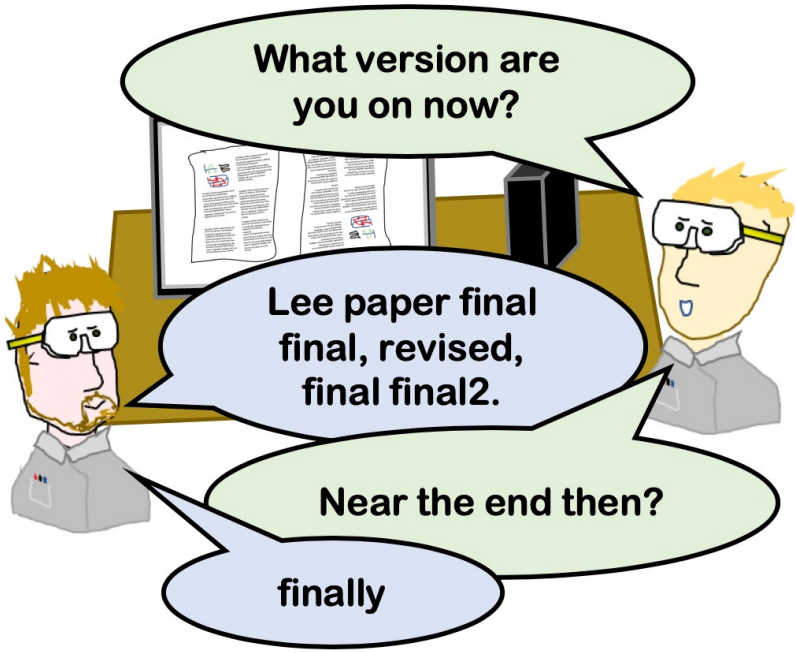
CS2410

Outlines

- **What is Git?**
  - **Version Control**
  - Git Introduction
- Basic Concept
  - Branch - merge, checkout
  - Local & Remote Repository - clone, push, pull
  - Adding Files to Git - add, commit
  - Open Source Issue - Fork & Pull Request
  - Conflict

Version Control

- **Version control** is a system that records changes to a file or set of files over time so that you can recall specific versions later.



Why do we need Git?

- What is the better way to do version control?
- Imagine you are working on a project with 10 people, how do you synch your files?

Outlines

- **What is Git?**
  - Version Control
  - **Git Introduction**
- Basic Concept
  - Branch - merge, checkout
  - Local & Remote Repository - clone, push, pull
  - Adding Files to Git - add, commit
  - Open Source Issue - Fork & Pull Request
  - Conflict

What does Git do?

- Easy access to different versions
- Compare changes over time
- See something is last modified by who
- Easy recovery when screwing things up

Git vs. GitHub vs. GitLab

Git	GitHub	GitLab
• Version Control System • A tool to manage your project	• GitHub is a code hosting platform for version control and collaboration • Based on Git • Free account limit to public repo (Before 2020) • Create private repo with free account after 2020	• GitLab is same to GitHub • Install GitLab on your own server very easy • Build-in CI/CD tool • Create private repo with free account • Five-user limit after 2023

Other Version Control Tools

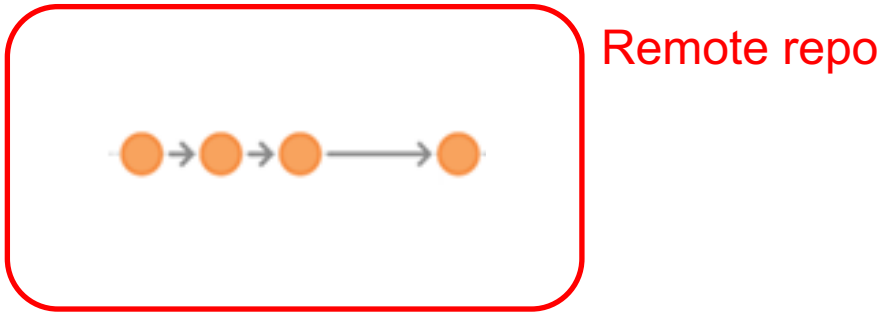
SVN

Outlines

- What is Git?
  - Version Control
  - Git Introduction
- **Basic Concept**
  - **Local & Remote Repository - clone**
  - Branch - merge, checkout
  - Update Files to Git - add, commit, push pull
  - Open Source Issue - Fork & Pull Request
  - Conflict
  - Rollback

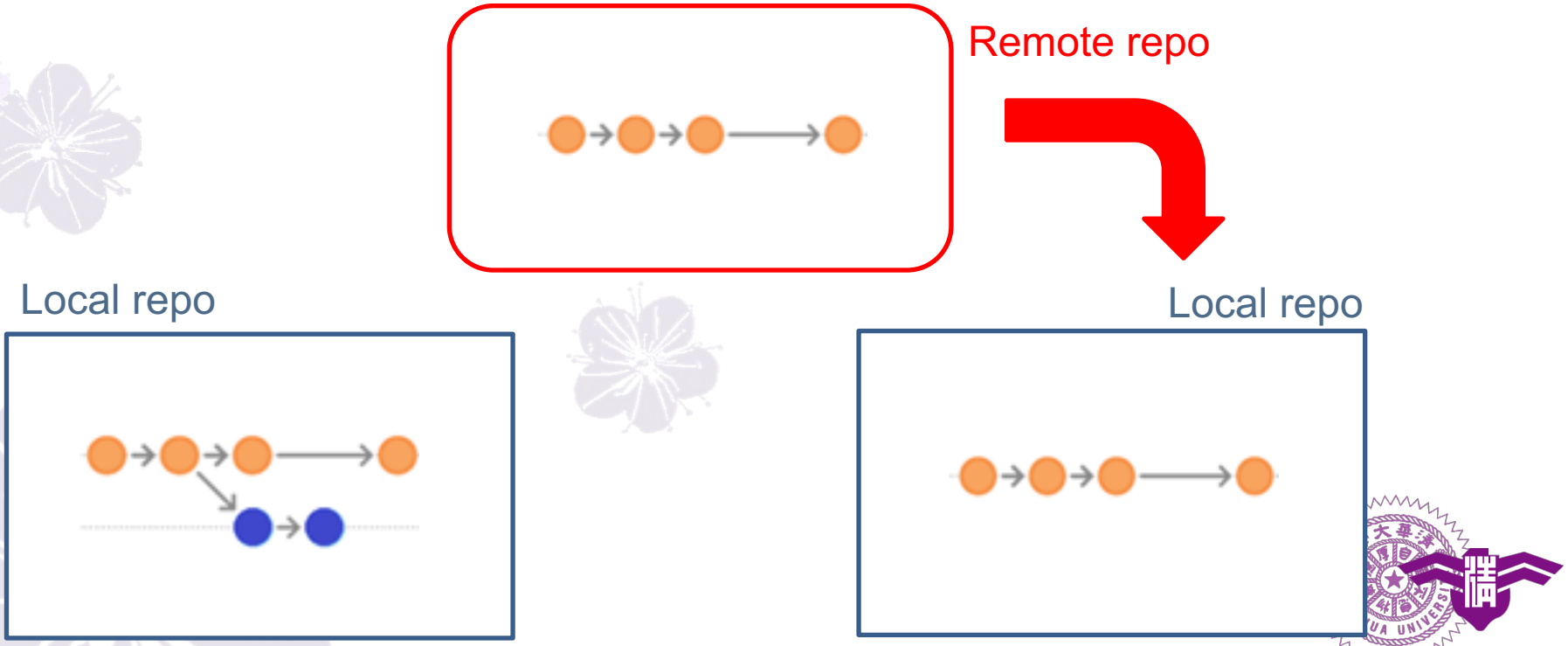
Local & Remote Repository

- **Repository** (repo) is the virtual storage of our data. It stores the branches and the versions of the project.



Local & Remote Repository

- We first **clone** from remote machine or initialize a local repository.



Command

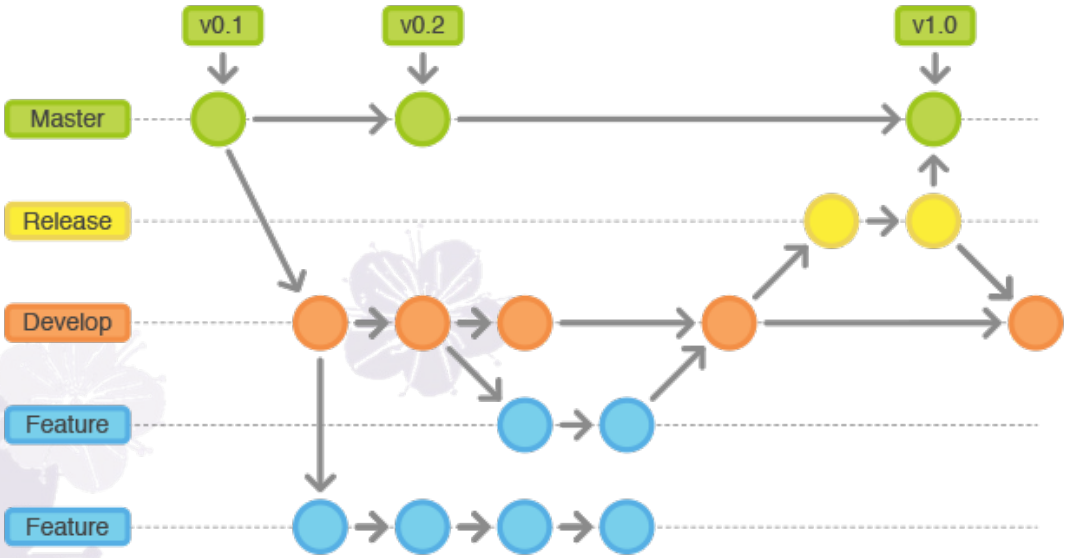
- git clone <url>

Outlines

- What is Git?
  - Version Control
  - Git Introduction
- **Basic Concept**
  - Local & Remote Repository - clone
  - **Branch - merge, checkout**
  - Update Files to Git - add, commit, push, pull
  - Open Source Issue - Fork & Pull Request
  - Conflict
  - Rollback

Branch

- **Branches** are where Git stores all the files and changes.
- Each node represents a version.



Branch Naming Convention

- **Master** : All official versions available to users are released on this master branch.
- **Dev** : The latest and most fully functional branch for daily development. If you want to be officially released, it will be merged back to the master branch.
- **Feature** : Some features are in the development phase.

Branch Naming Convention

- **Release** : Before formally being merged back to master, release the version that is regularly available online.

  (Release is usually branched out from the dev branch, and after the pre-release, it will be merged back to master and dev.)
- **Bug** : After the official release, this branch is used to repair bug on the program.

  When the repair is complete, it is merged back to master and dev.

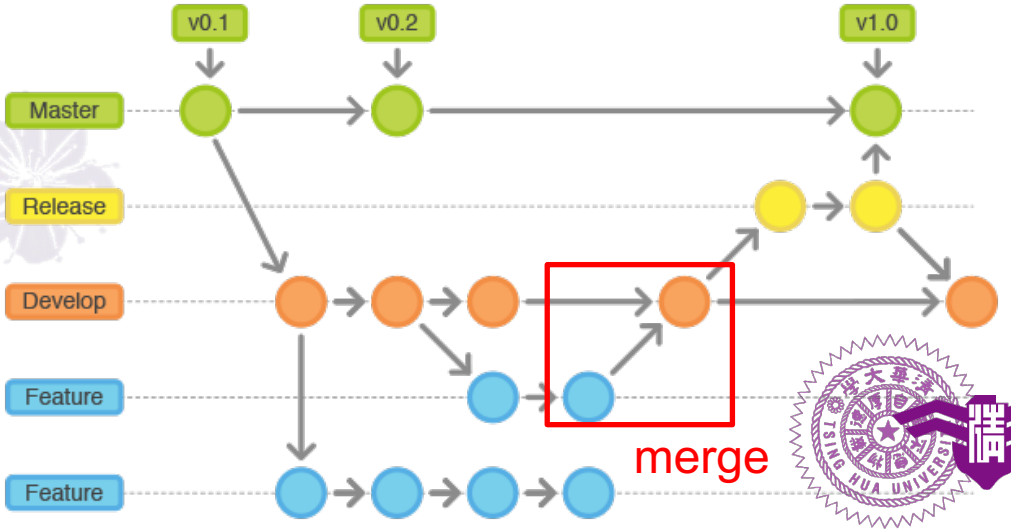
Branch

- Everyone works on their own branch, and the branches are then **merged** back to the shared branches.
- Use **checkout** to move between versions and branches.

Tip:

When we checkout an old version of current branch, remember to create a new branch for it!

The original branch will still stay at the latest version of that branch.



Command

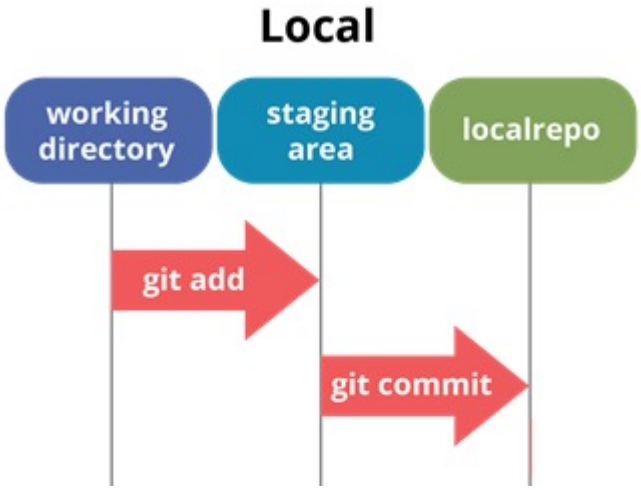
- git checkout –b <branch>: create and switch to branch
- git checkout <branch>: switch to existed branch
- git merge <branch>: merge branch to now working branch

Outlines

- What is Git?
  - Version Control
  - Git Introduction
- **Basic Concept**
  - Local & Remote Repository - clone
  - Branch - merge, checkout
  - **Update Files to Git - add, commit, push, pull**
  - Open Source Issue - Fork & Pull Request
  - Conflict
  - Rollback

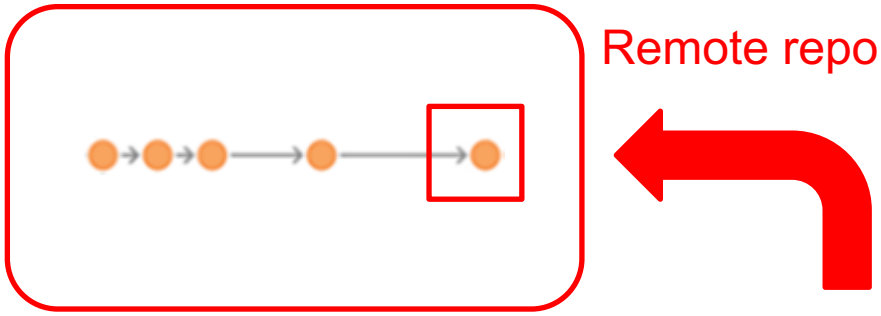
Update Files to Git

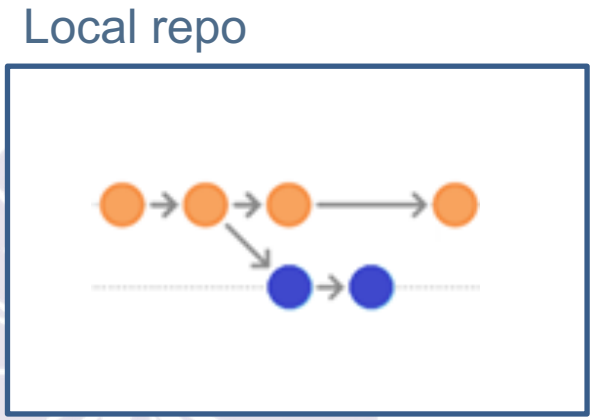
- When we want to state our changes, we first **Adding** them.
- Then **Committing** them to enter the git system.
- Each commit forms a node on current branch.
- Final, we **Pushing** them to remote repo.

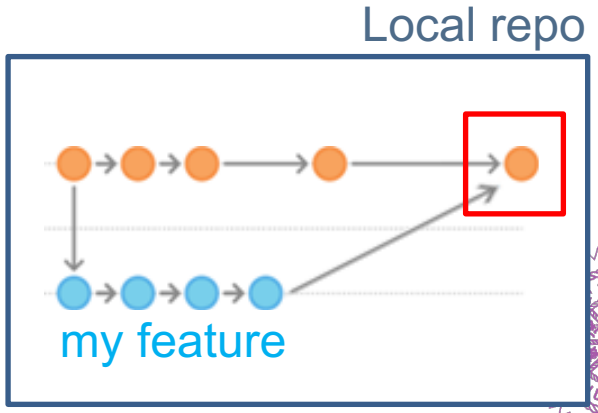


Update Files to Git

- After we change the content of the local branches, we **push** to update changes of to the remote branch.

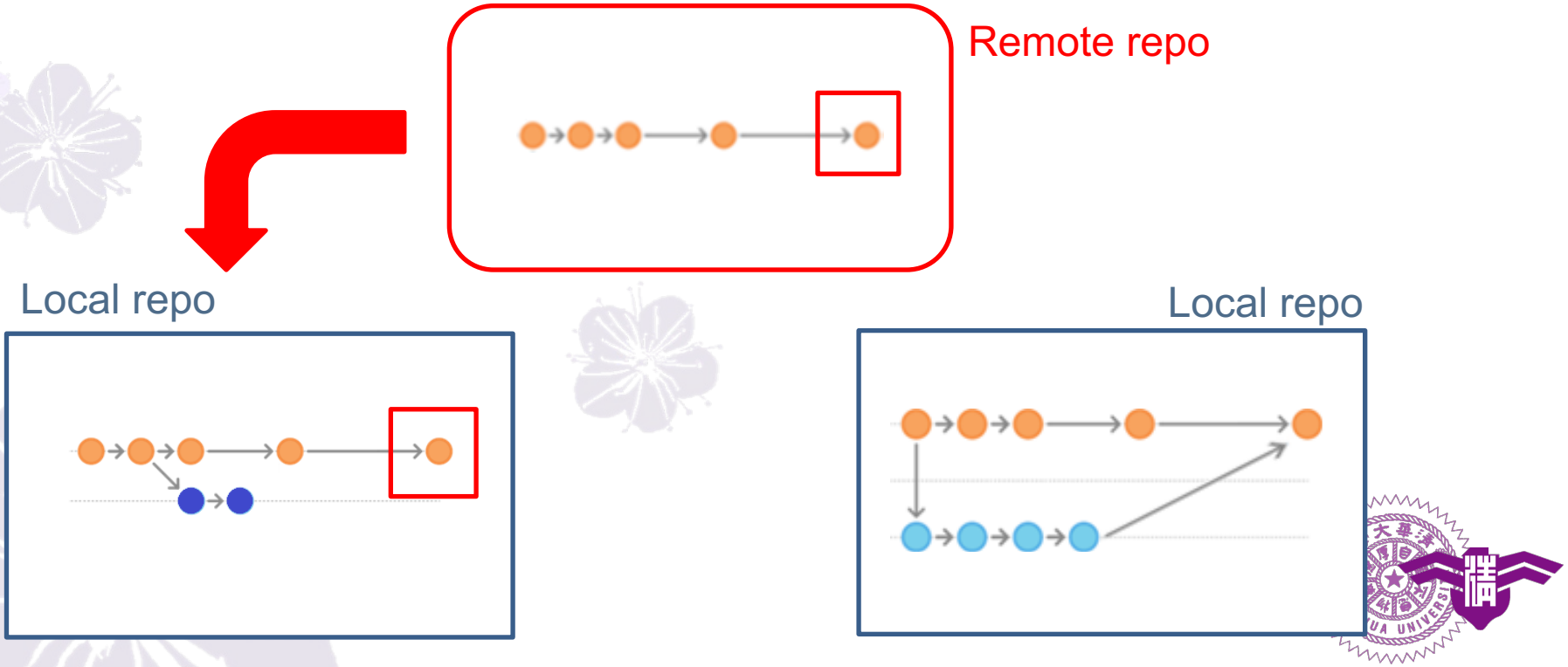






Update Files to Git

- Using **pull** command to sync with remote repo before starting to work.



Command

- git add <files>
- git commit –m “<commit message>”
- git push
- git pull

Tip:

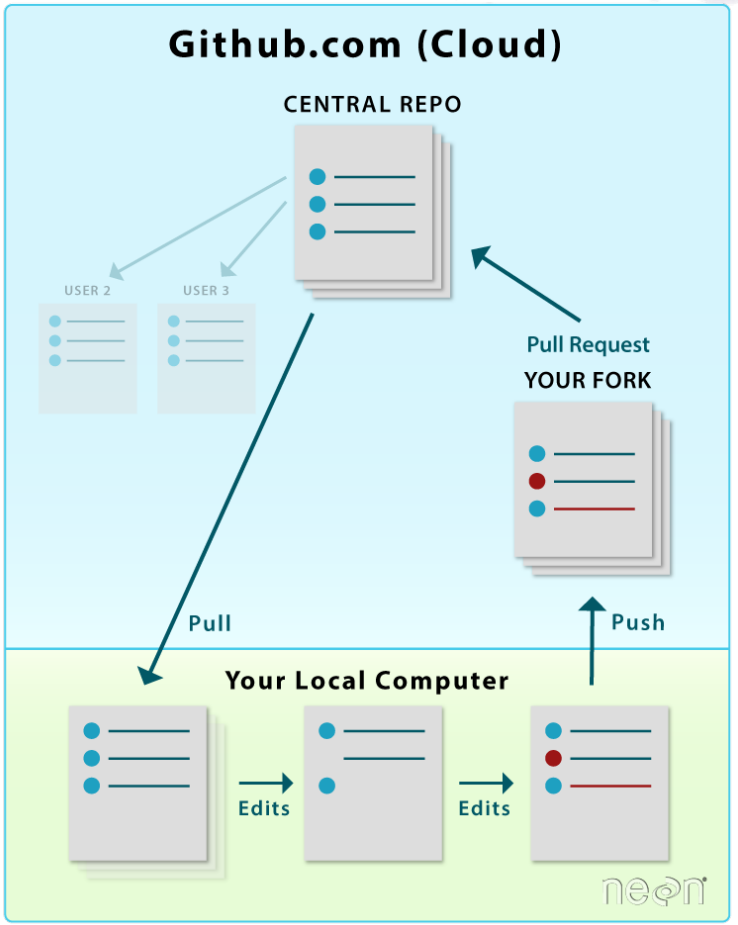
When writing “commit message”, please focus on the purpose of this change, such as what is the difference from old version, what you had fixed, etc..

Outlines

- What is Git?
  - Version Control
  - Git Introduction
- **Basic Concept**
  - Local & Remote Repository - clone
  - Branch - merge, checkout
  - Update Files to Git - add, commit, push, pull
  - **Open Source Issue - Fork & Pull Request**
  - Conflict
  - Rollback

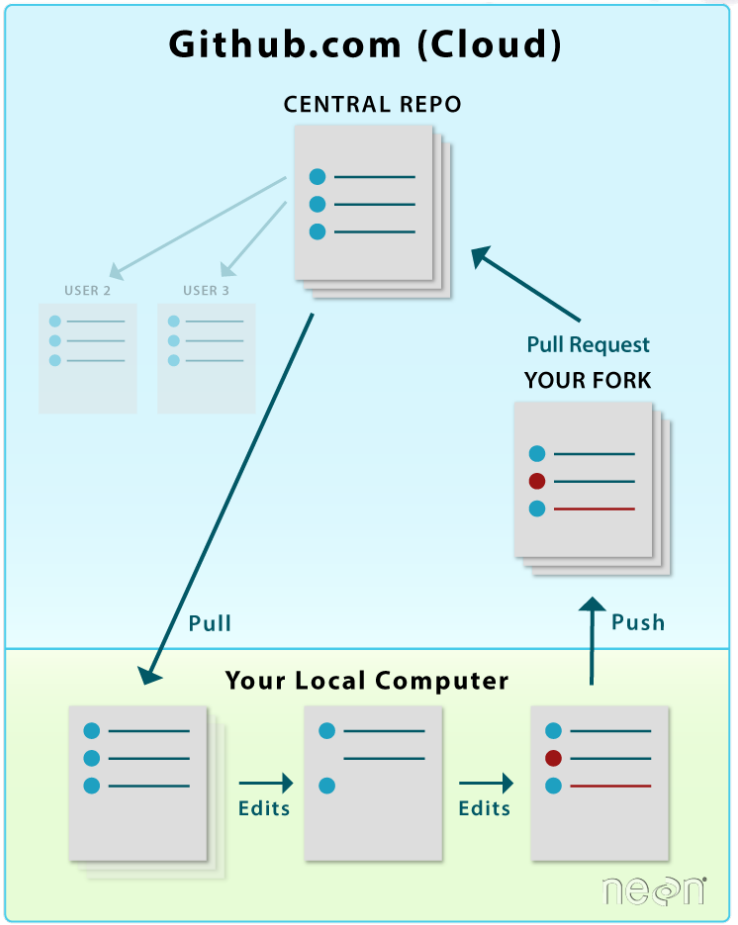
Open-Source Issue

- Open-source projects can be easily polluted if they allow everyone to contribute to project.



Open-Source Issue

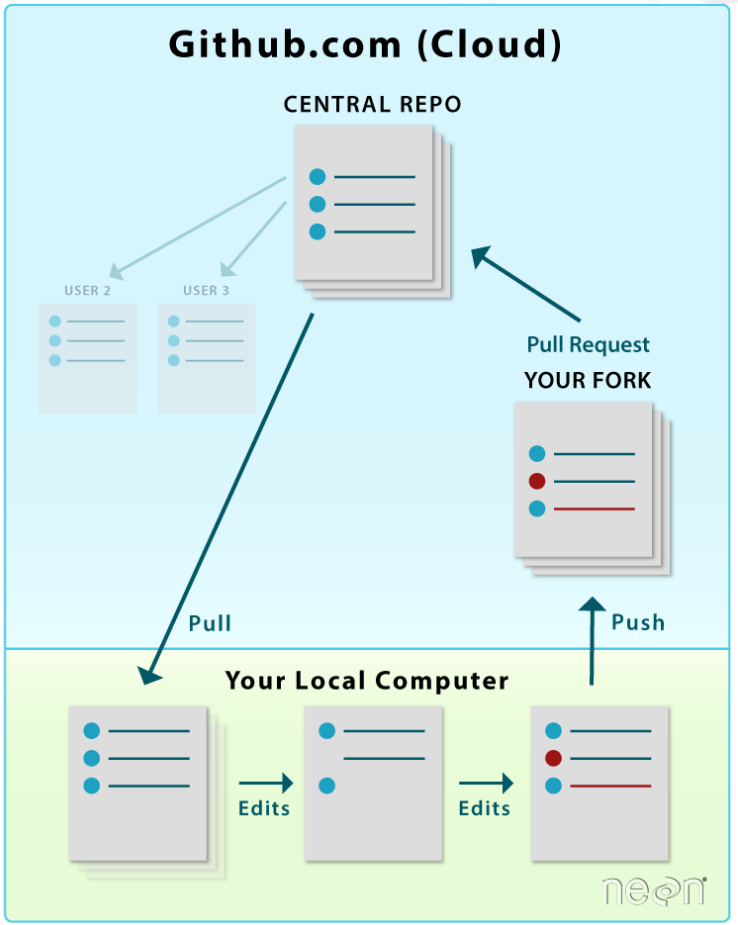
- A better way doing this is to **fork** the whole project to a new remote repository, where the data can be modified as we want.



We do these with GitLab, not command lines!

Open-Source Issue

- After coding, make a **pull request** to ask if the project team are willing to merge the modified code back to their project.



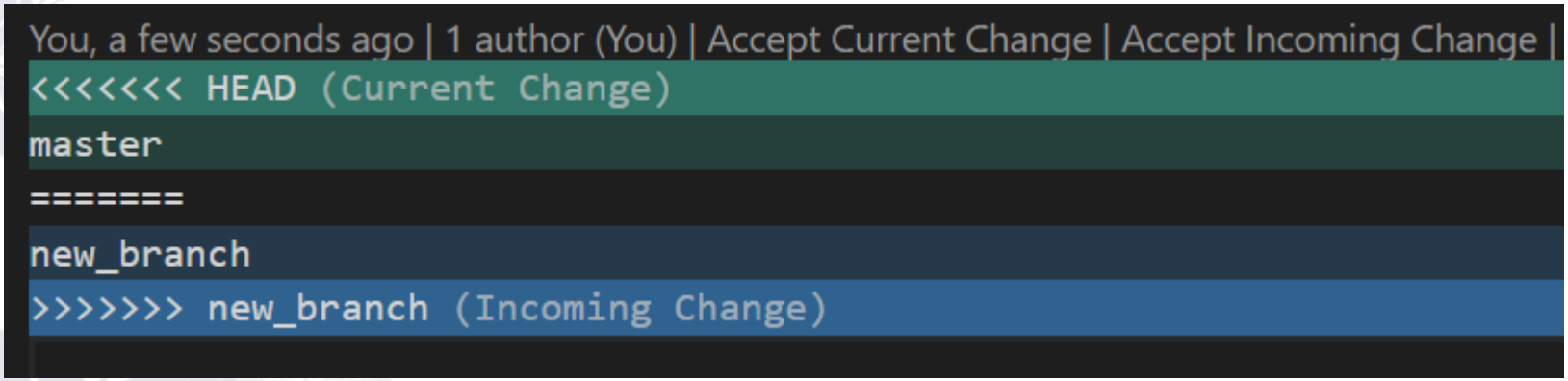
We do these with GitLab, not command lines!

Outlines

- What is Git?
  - Version Control
  - Git Introduction
- **Basic Concept**
  - Local & Remote Repository - clone
  - Branch - merge, checkout
  - Update Files to Git - add, commit, push, pull
  - Open Source Issue - Fork & Pull Request
  - **Conflict**
  - Rollback

Conflict

- **Conflicts** may occur when we try to merge the branches or pull from remote repo. If this happens, just solve it manually.



This tells you where the conflict is, delete them and keep what you want.

Outlines

- What is Git?
  - Version Control
  - Git Introduction
- **Basic Concept**
  - Local & Remote Repository - clone
  - Branch - merge, checkout
  - Update Files to Git - add, commit, push, pull
  - Open Source Issue - Fork & Pull Request
  - Conflict
  - **Rollback**

Rollback

- When we unexpectedly ruin our current project, sometimes we need to **rollback** to old version.
- Print all of our commit **logs** and remember the wanted version's commit id.
- Then **reset** to old commit.

Command

- git log
- git reset <commit id>

```
(base) licairong@licairongdeMacBook-Pro projected-texture % git log
commit 0a9bd0d8397a5a967fd7179d159e0419ba6c8574 (HEAD -> main, origin/main, origin/HEAD)
Author: 107062117
Date:   Tue Nov 23 16:16:23 2021 +0800
```

Commit id

實際情境模擬

小智、小剛、小美三人決定開發一款男女老幼都愛不釋手的遊戲，於是在正式開始製作之前，先做好分工，並決定使用 gitlab 幫助版本控管，讓這次project能更加簡便順利。

小智

角色設計

小剛

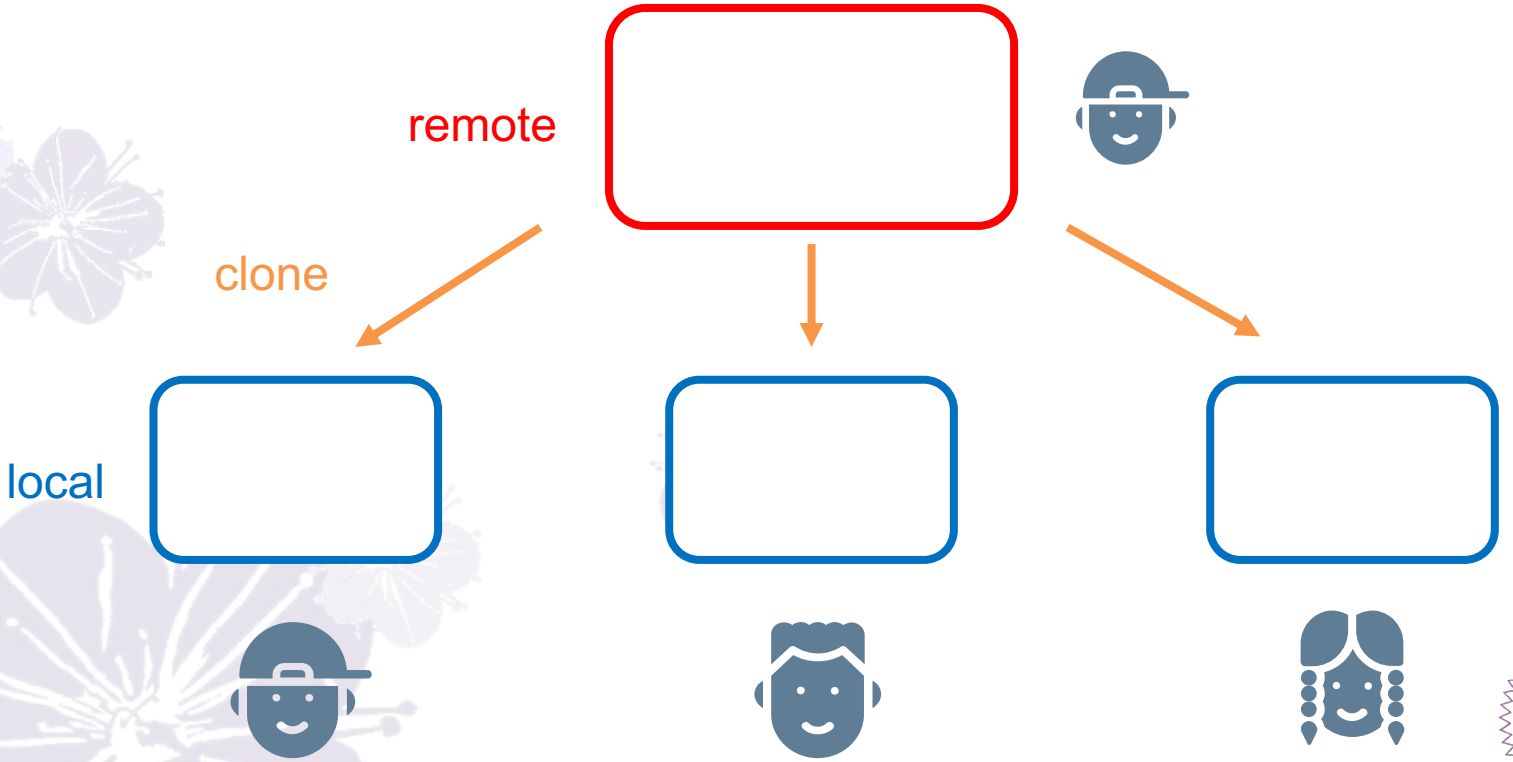
場景、UI介面設計

小美

關卡設計

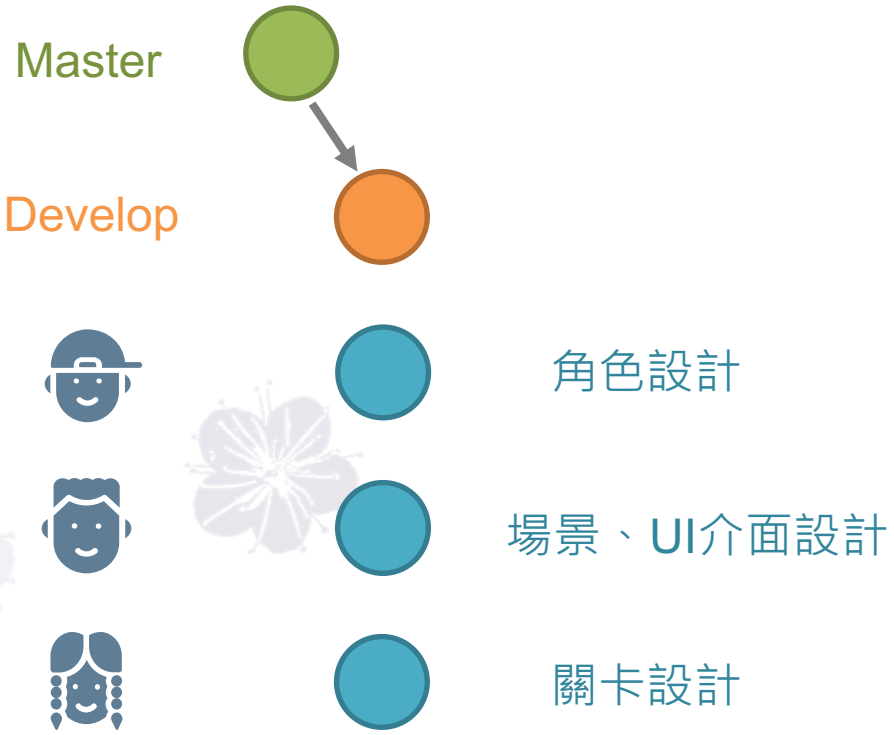
實際情境模擬

首先，小智先在gitlab上創建好project，三人再分別將project clone到自己的電腦上，準備開始進行遊戲的開發。



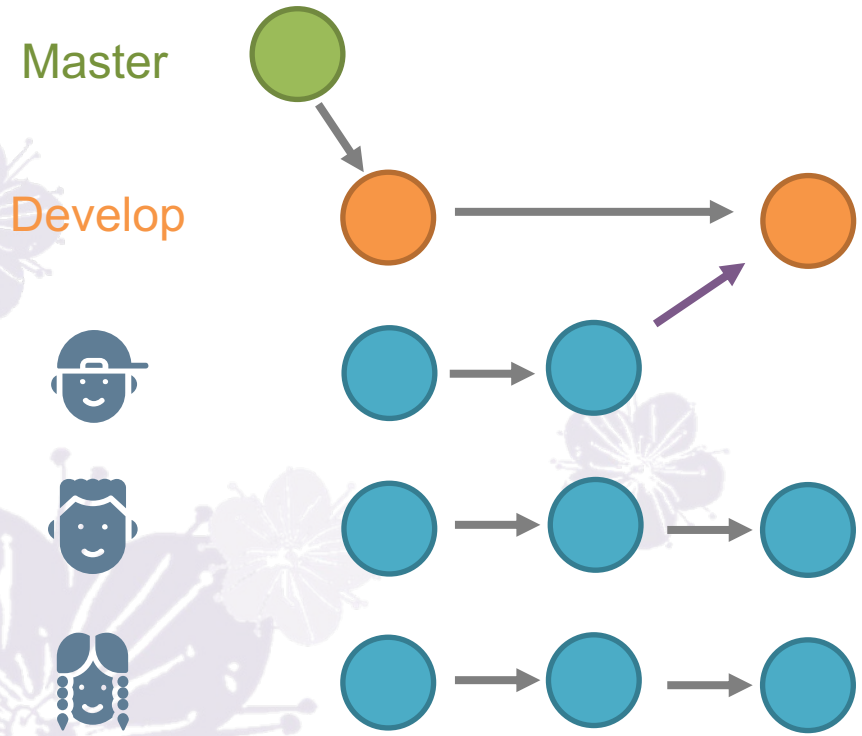
實際情境模擬

接著三人就照著之前的分工，分別建立自己負責的branch，在自己的branch上將工作完成。



實際情境模擬

小智先完成了他的工作，因此他先將自己做的東西merge回develop，其他人繼續完成工作。

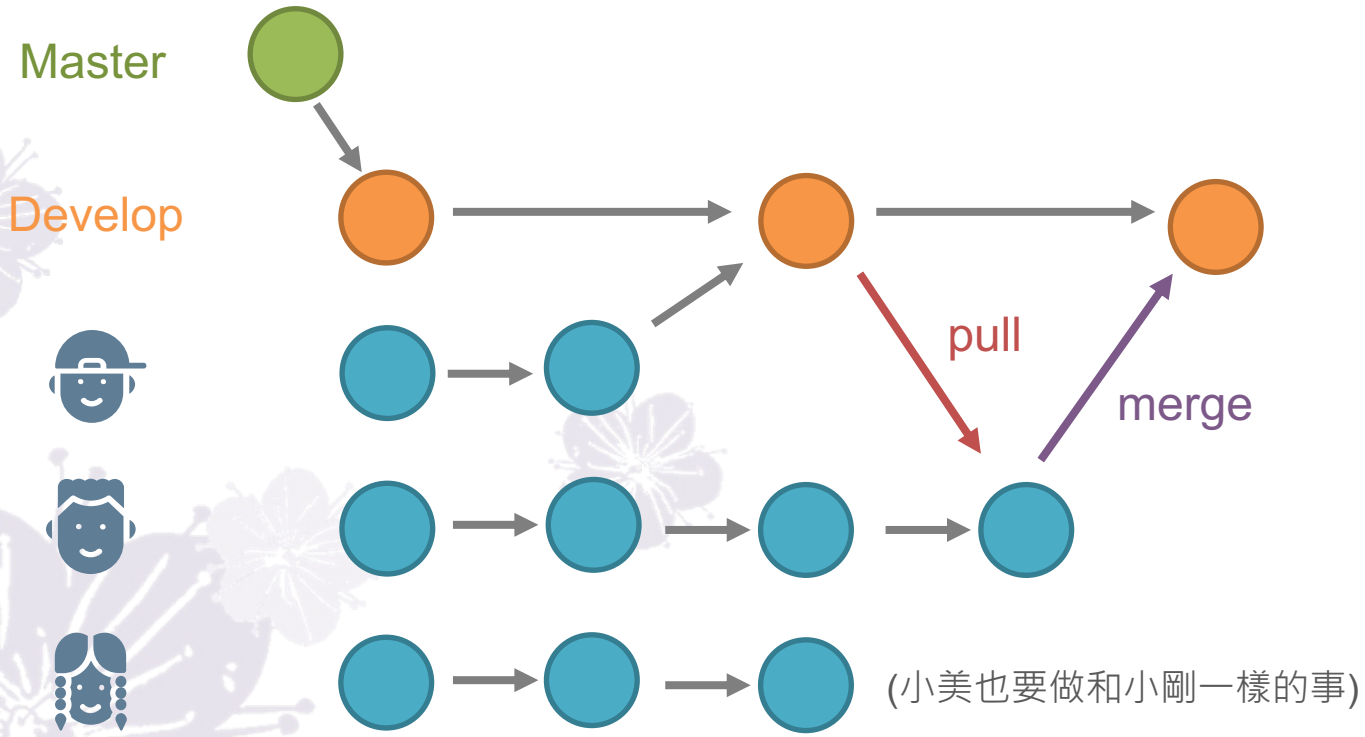


Merge簡單流程：

1. 先add、commit自己的branch
2. Checkout到develop的branch
3. Merge小智完成的branch
4. Push merge好的branch到remote repo

實際情境模擬

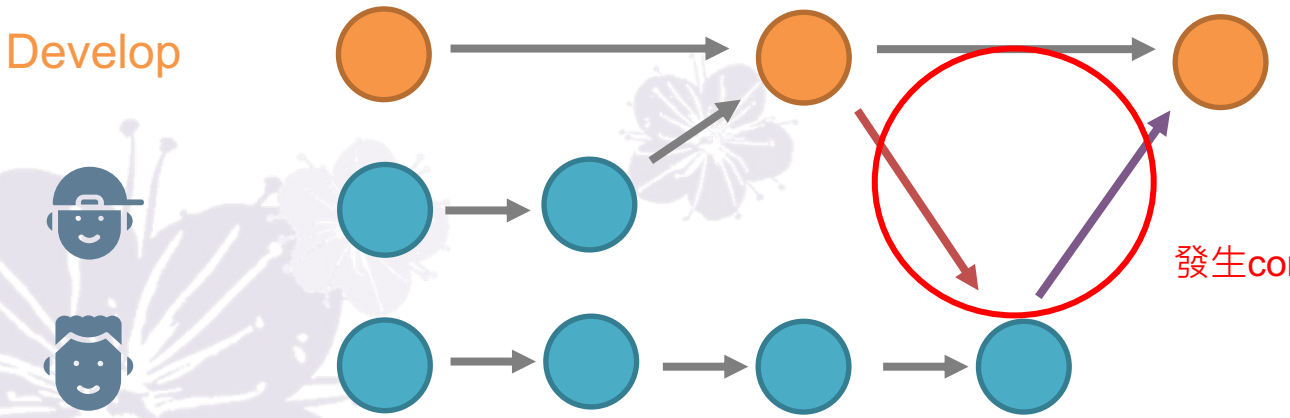
之後小剛和小美也都陸續完成自己的工作，但因為develop已經有小智更新過的東西，所以此時他們在add、commit後，還要先pull目前最新的版本，進行合併，再merge回develop。



實際情境模擬

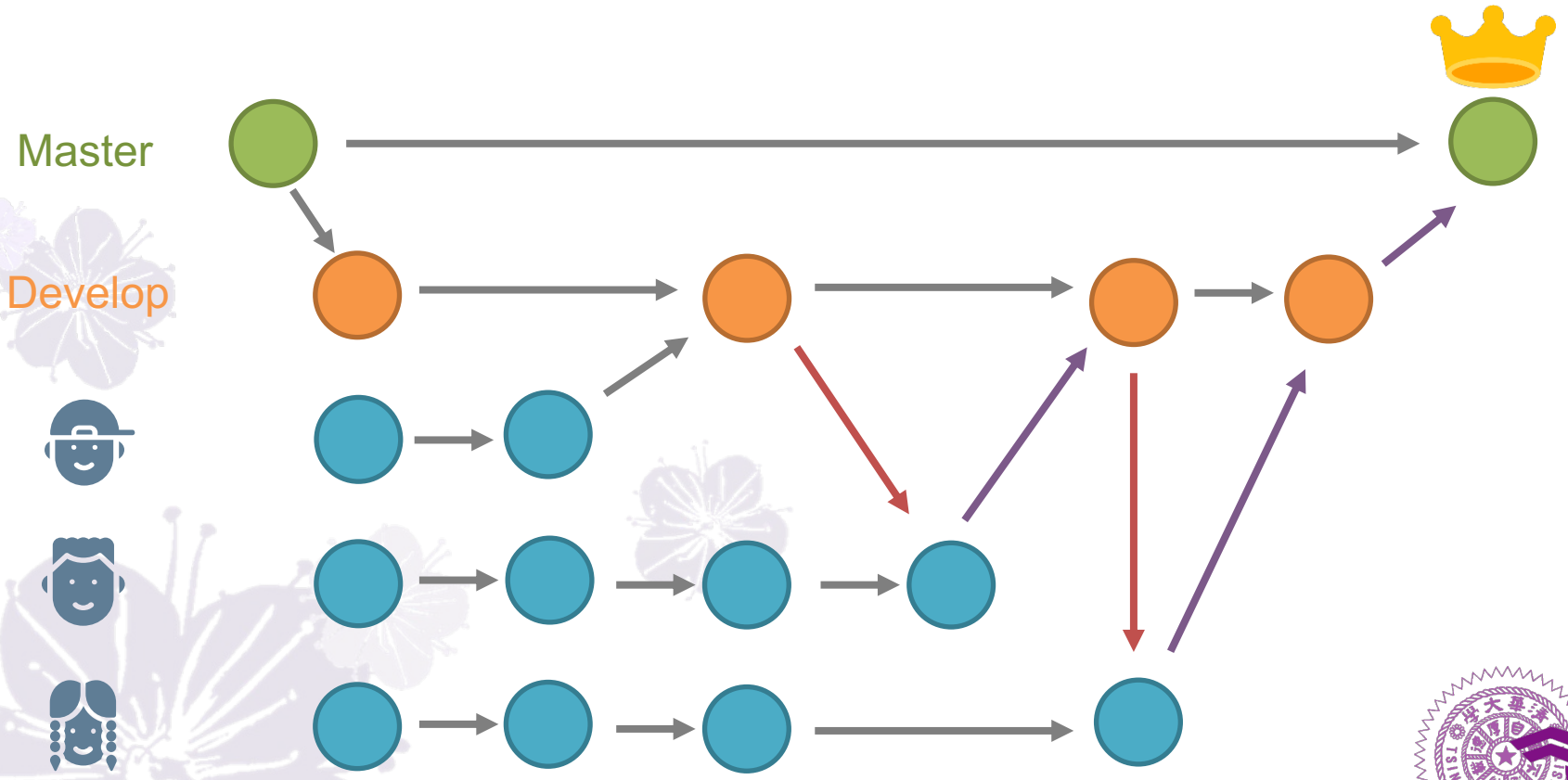
在做pull或是merge的時候，可能會因為兩個人同時改到了相同的地方，所以產生了conflict。

例如小智制作角色時，先隨便建立場景，因此小剛要將做完的場景和小智已經丟上去的東西合併時，就會不知道要留小智的場景還是小剛的場景。此時因為小剛才是製作場景的人，所以在解除conflict時，就會選擇小剛的版本，而不是小智的版本。



實際情境模擬

最後，小剛小美完成合併後，做最後的檢查，就可以合併回master囉！以上就是簡單的實際情境模擬。



Git Tutorials

- Git for Professionals Tutorial: https://www.youtube.com/watch?v=Uszj_k0DGsg&t=1115s
- Top 20 Git Commands With Examples: https://dzone.com/articles/top-20-git-commands-with-examples
- 連猴子都能懂的Git入門指南
- Search for ‘Git Tutorial’