

Tóm tắt những lệnh git đã học và sẽ áp dụng trong bài tập này:

\$ git push

\$ git commit -m "commit message"

\$ git checkout | git checkout -b | git checkout <branch-name>^ | git checkout <branch-name>~2 | git checkout HEAD^ | git checkout HEAD~2

\$ git merge <branch-name>

\$ git rebase HEAD

\$ git rebase -i HEAD

\$ git reset HEAD^

\$ git revert HEAD -m 1

\$ git cherry-pick C1 C2

\$ git tag v0.1 C1

\$ git archive --output=archive-v.0.1.zip v.0.1

Những điều cần lưu ý khi làm bài:

- Mỗi commit sẽ được cung cấp trong folder "git commits", mỗi tên thư mục commit sẽ tương ứng với message commit và tên commit trong đề bài (VD: thư mục C1 sẽ chứa các file trong commit C1, mn chỉ cần copy vào source hiện tại và commit với message là C1).
- Với mỗi bài tập, hình INPUT sẽ là điều kiện tiền đề để thực hiện bài tập, kết quả cần đạt được sẽ được mô tả ở hình GOAL.
- Trước mỗi bài tập đều sẽ có phần hướng dẫn thực hiện, nhớ đọc kỹ nhé.
- Trong bài này, mình đã dùng phần mềm GitUp để trực quan hóa lịch sử git, bên cạnh đó, mn có thể dùng lệnh git để làm việc này với cú pháp:

```
$ git log --all --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit
```

Bỏ lệnh này vào alias để tiện cho việc sử dụng, cú pháp như sau:

```
$ git config --global alias.lg "log --all --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit"
```

Sau khi chạy, ta có thể sử dụng với lệnh git ngắn gọn: \$ git lg

BT 1:

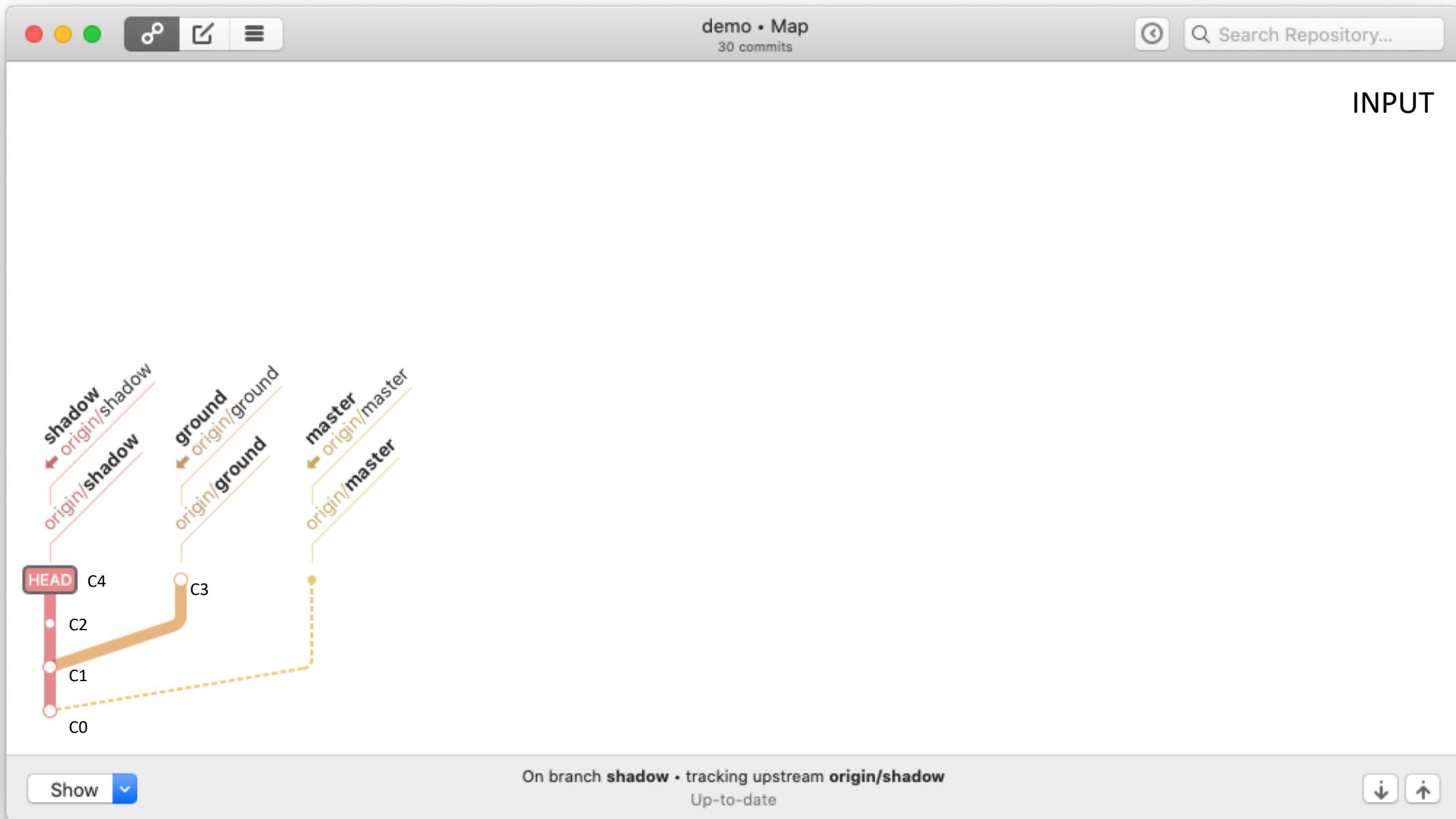
- Tạo 1 repository trên github, folder source code.
- Sử dụng ‘git remote add origin’ để thêm remote origin cho folder source.
- Tạo nhánh master, push commit C0

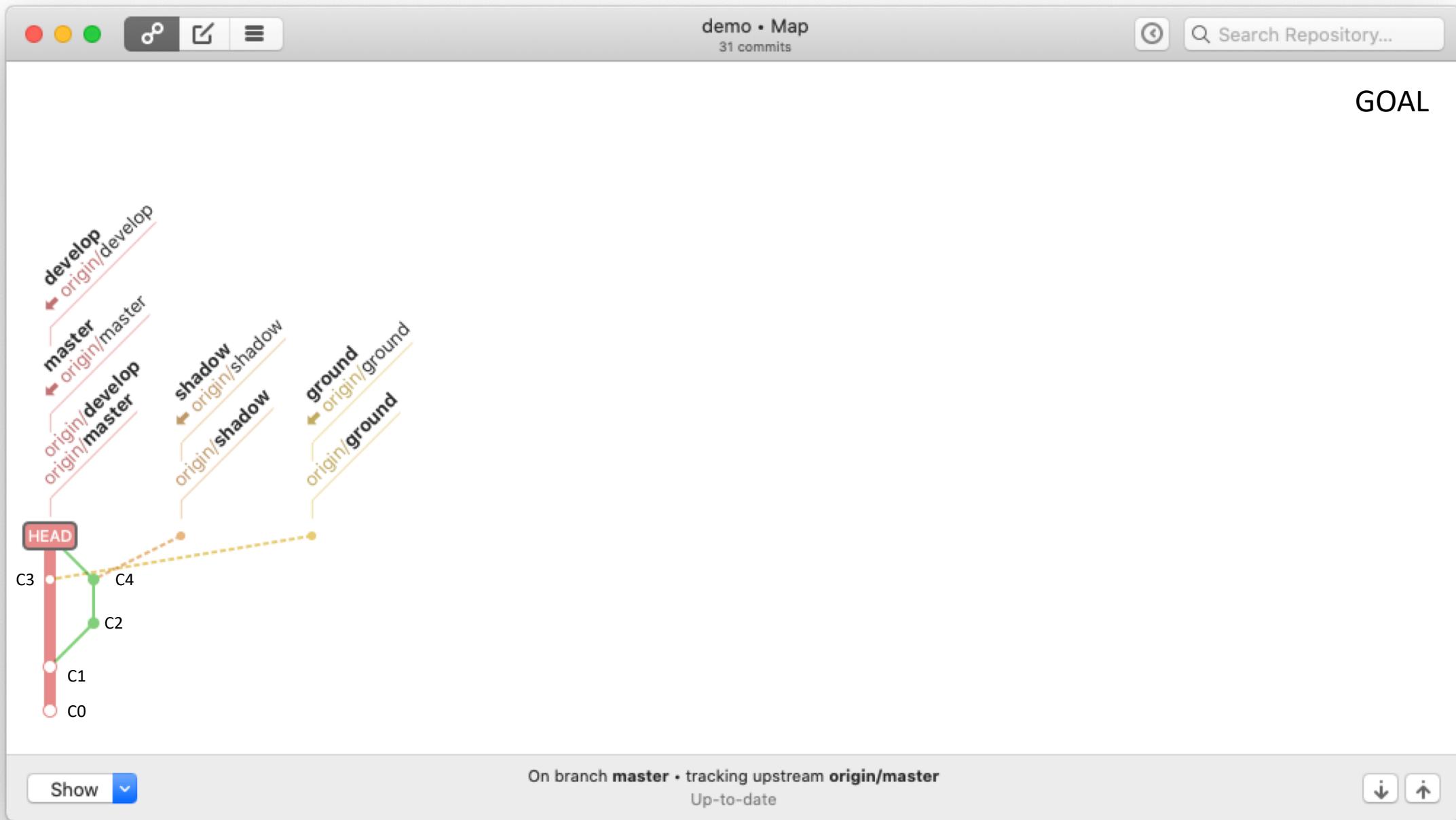
Tưởng tượng rằng project này đang có nhiều hơn 1 người code, do vậy bạn cần mô phỏng quá trình coding của nhiều người (nhiều nhánh)

- Tạo nhánh ground và push commit C1
 - Tạo nhánh shadow và push commit C2
 - Trở lại nhánh ground và push commit C3
 - Trở lại nhánh shadow và push commit C4
-

Yêu cầu:

- Tạo nhánh develop và merge 2 nhánh ground, shadow vào nhánh develop
- Merge nhánh develop vào nhánh master





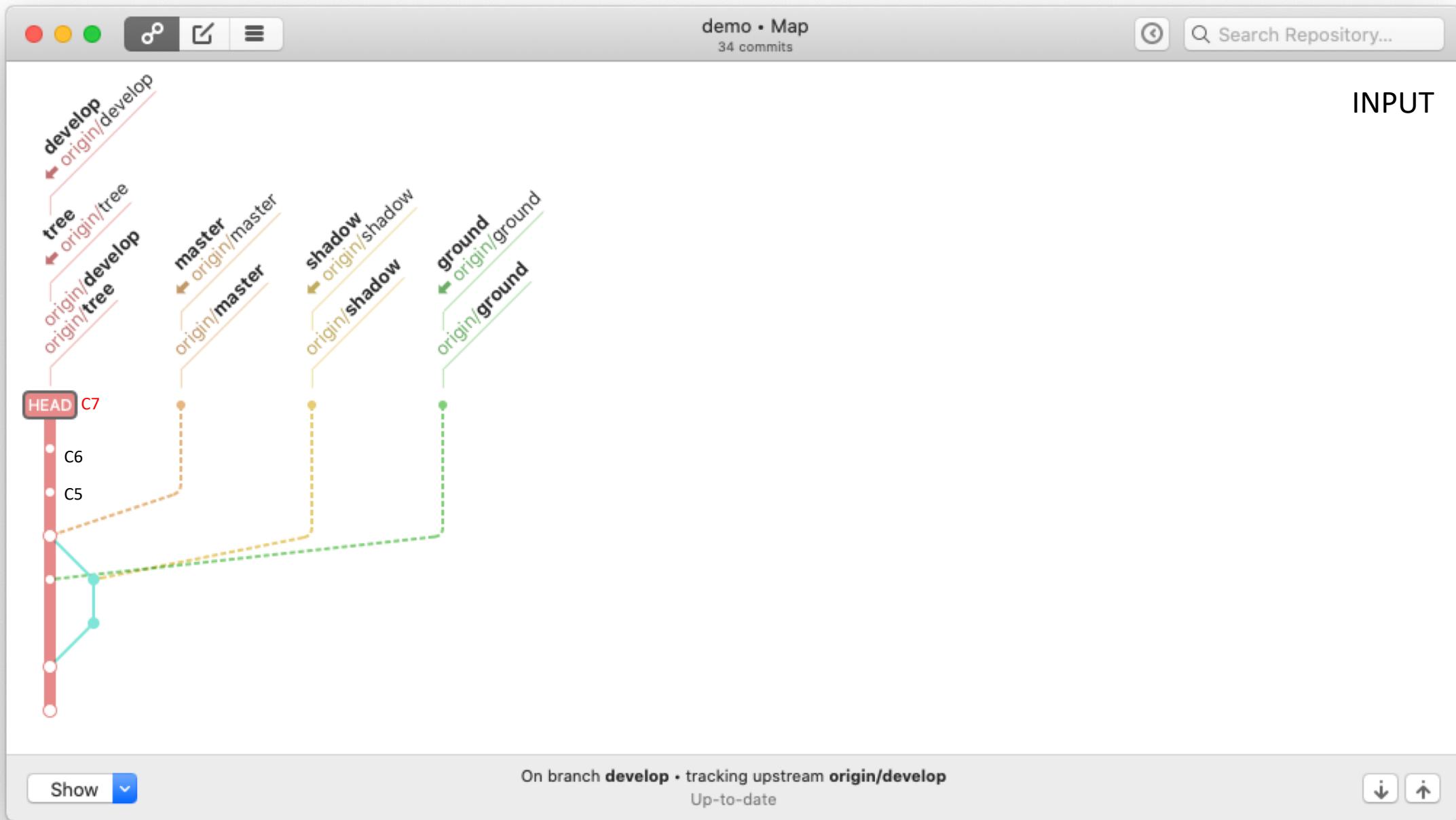
BT 2:

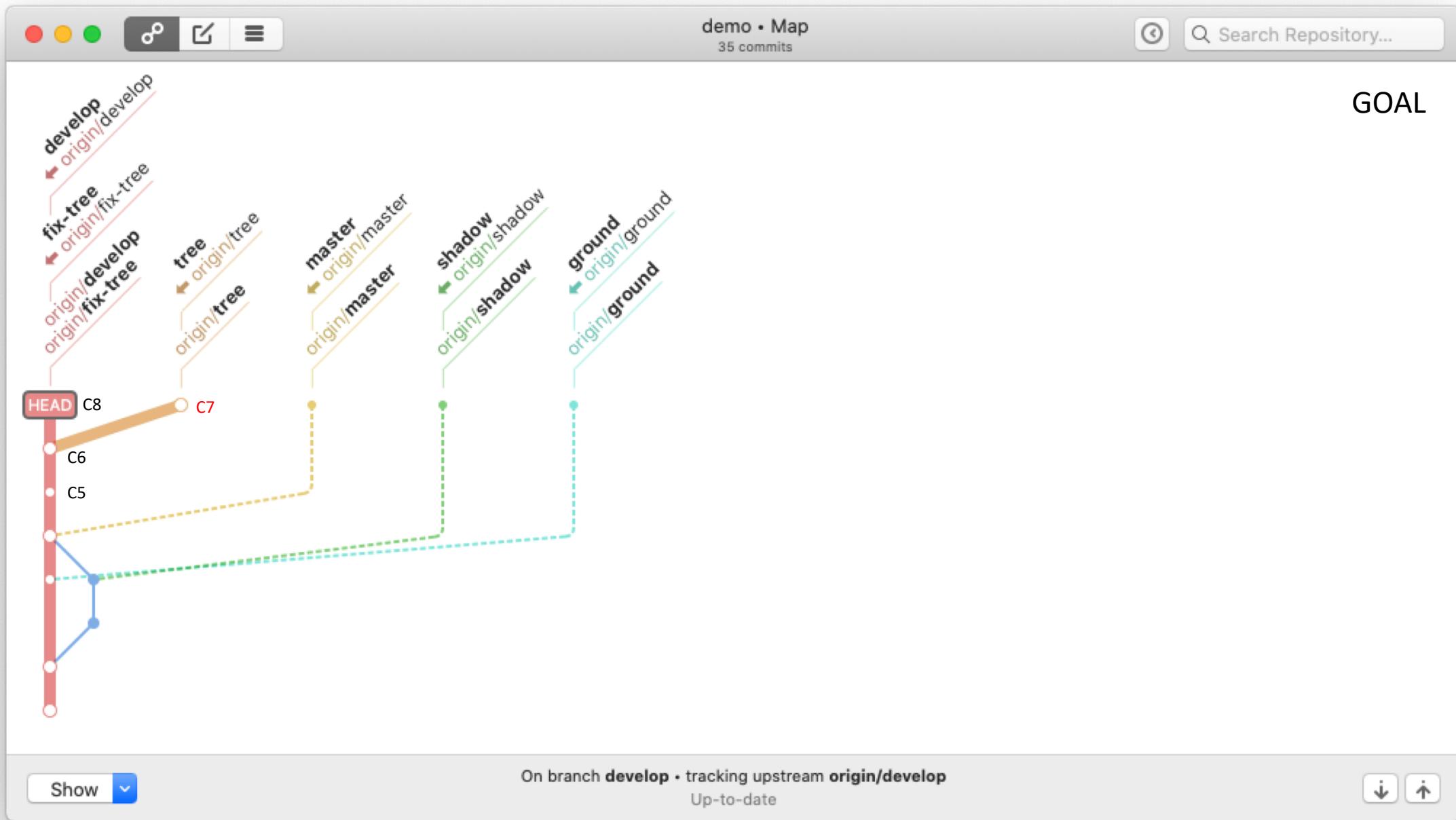
- Tại nhánh develop tạo nhánh tree.
- Push các commit C5, C6, C7.
- Merge nhánh tree vào nhánh develop.

Sau khi testing, phát hiện ra rằng commit C7 có BUG và cần loại bỏ để tiếp tục code

Yêu cầu:

- Loại bỏ commit C7 bằng cách tạo thêm 1 nhánh fix-tree tại commit C6
- Trên nhánh fix-tree, push commit C8 và đưa nhánh develop về vị trí commit C8





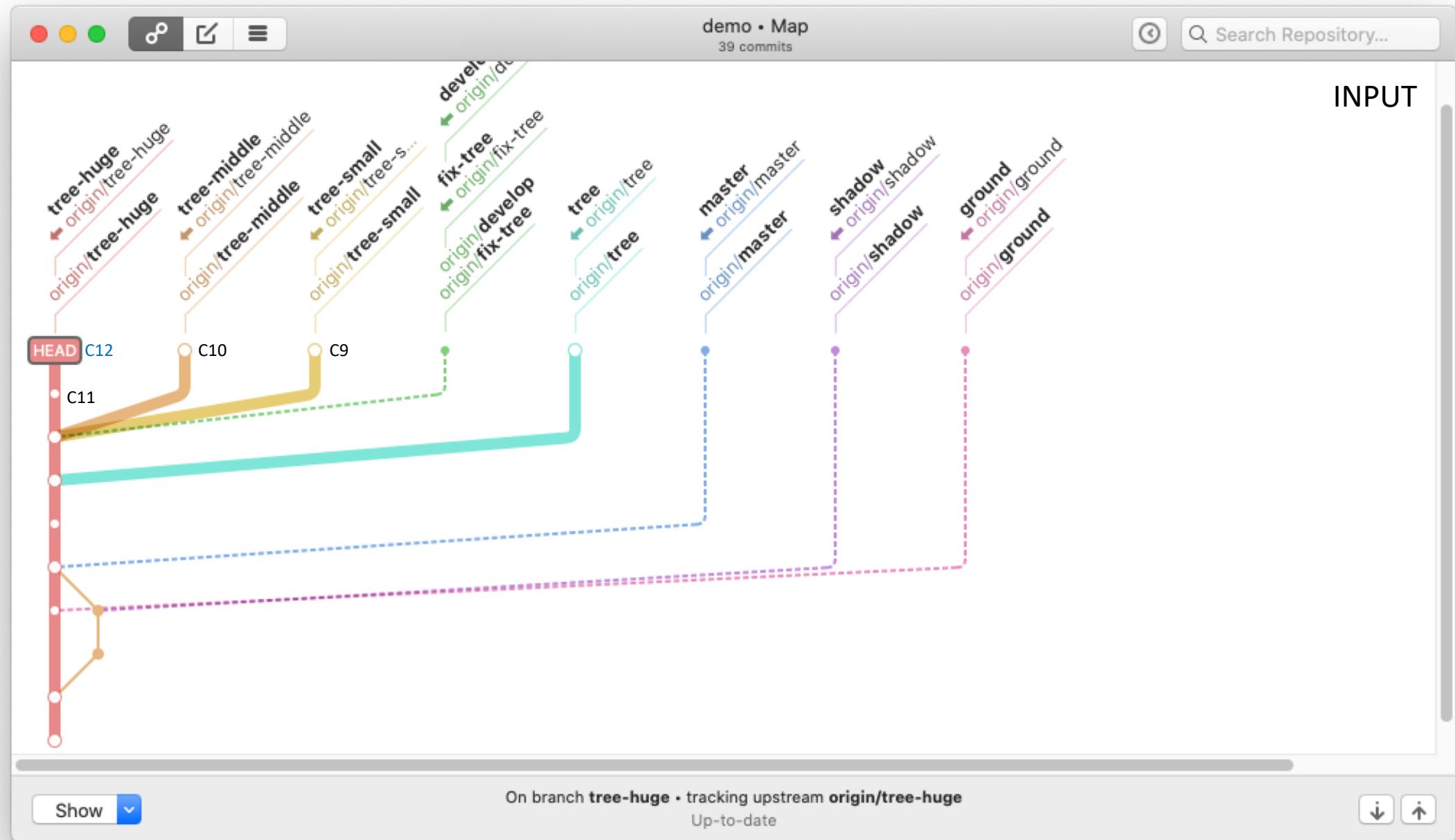
BT 3:

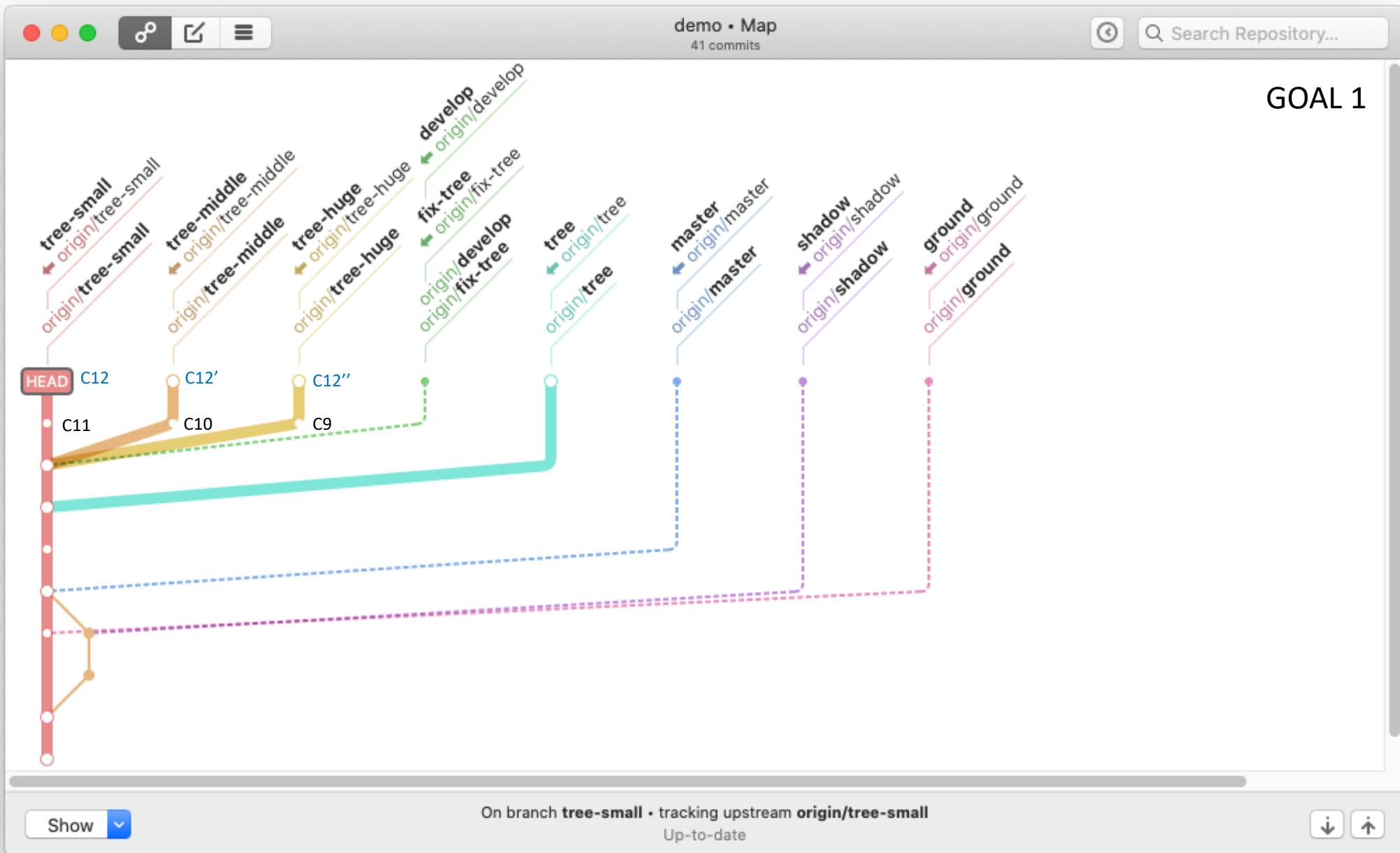
- Tại nhánh develop tạo nhánh tree-small và push commit C9.
- Trở lại nhánh develop, tạo nhánh tree-middle và push commit C10.
- Trở lại nhánh develop, tạo nhánh tree-huge và push commit C11, C12.

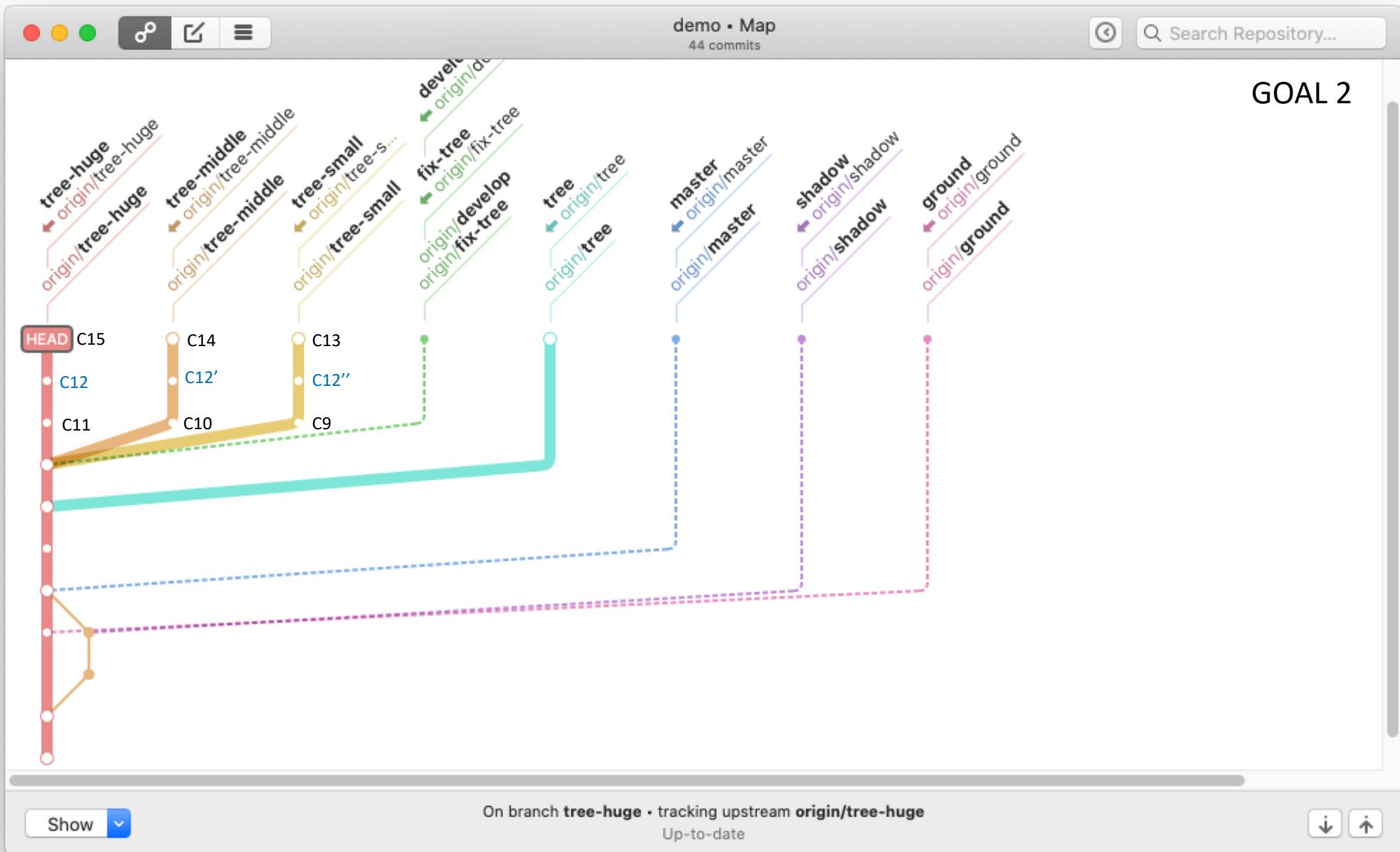
Commit C12 có chứa đoạn code có thể áp dụng cho cả 3 nhánh trên và có thể tránh được conflict sau này.

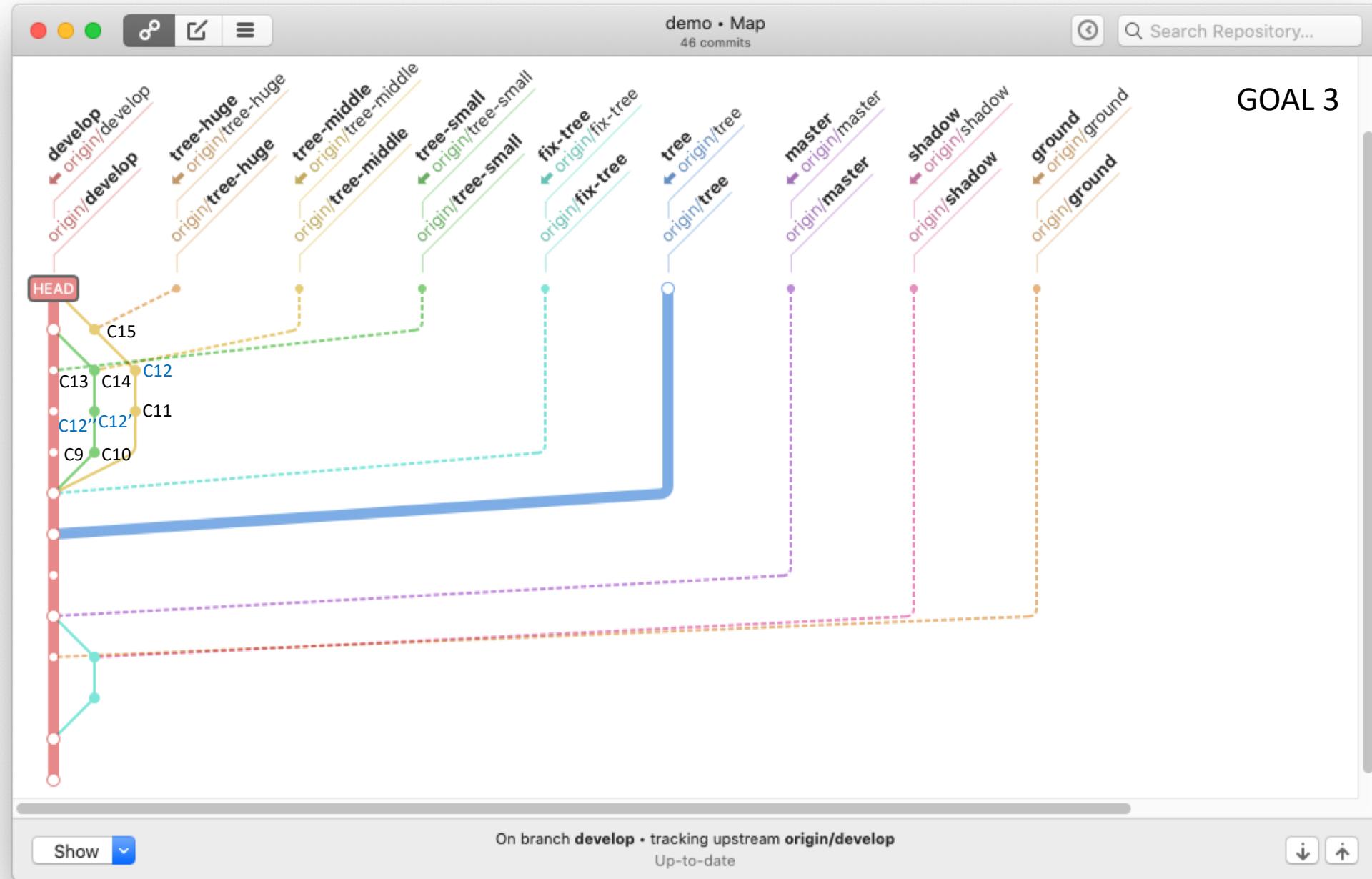
Yêu cầu:

- Áp dụng commit C12 vào cho nhánh tree-middle và tree-small
- Tiếp tục push tiếp các commit C13, C14, C15 để hoàn thiện tính năng
- Merge cả 3 nhánh vào nhánh develop









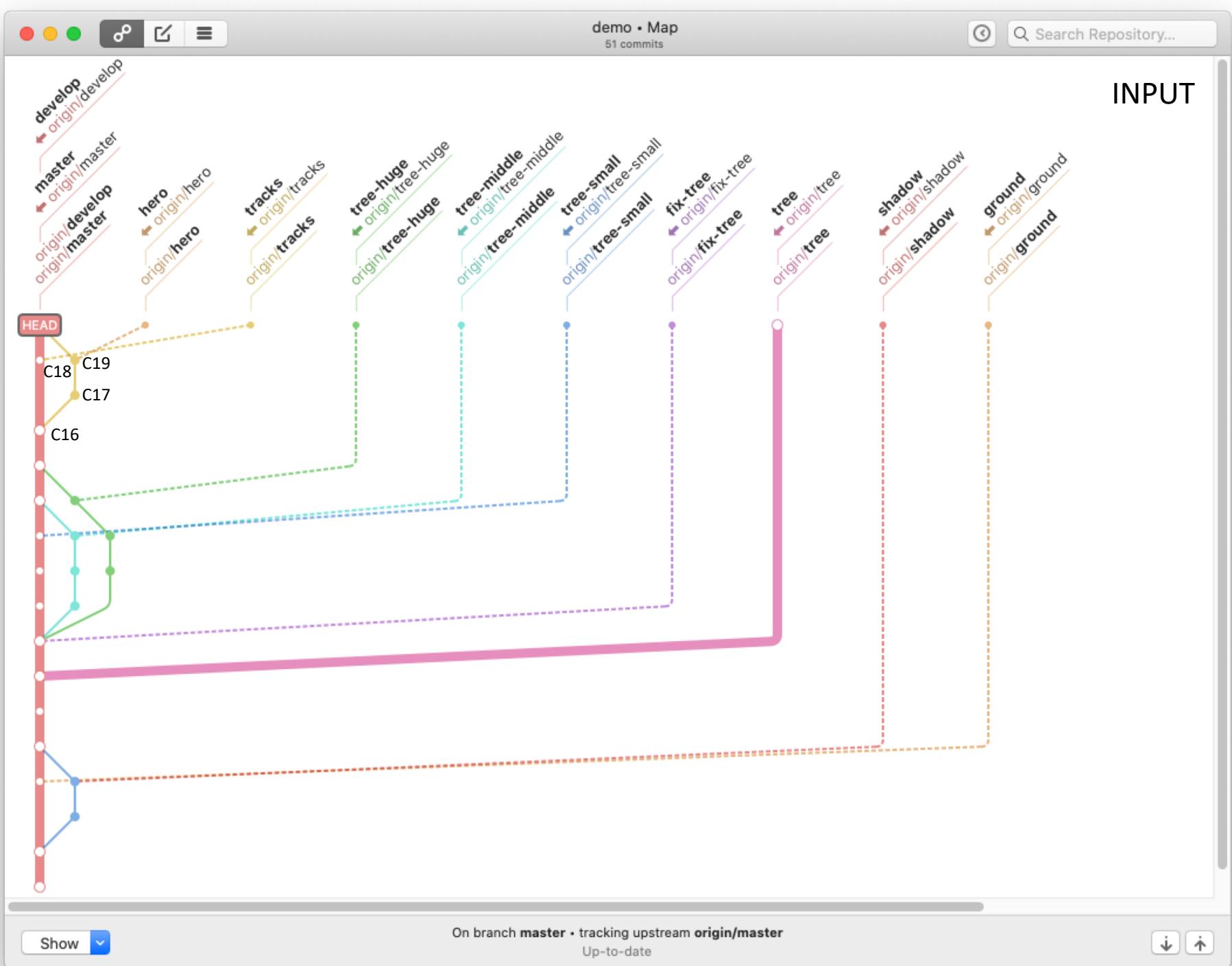
BT 4:

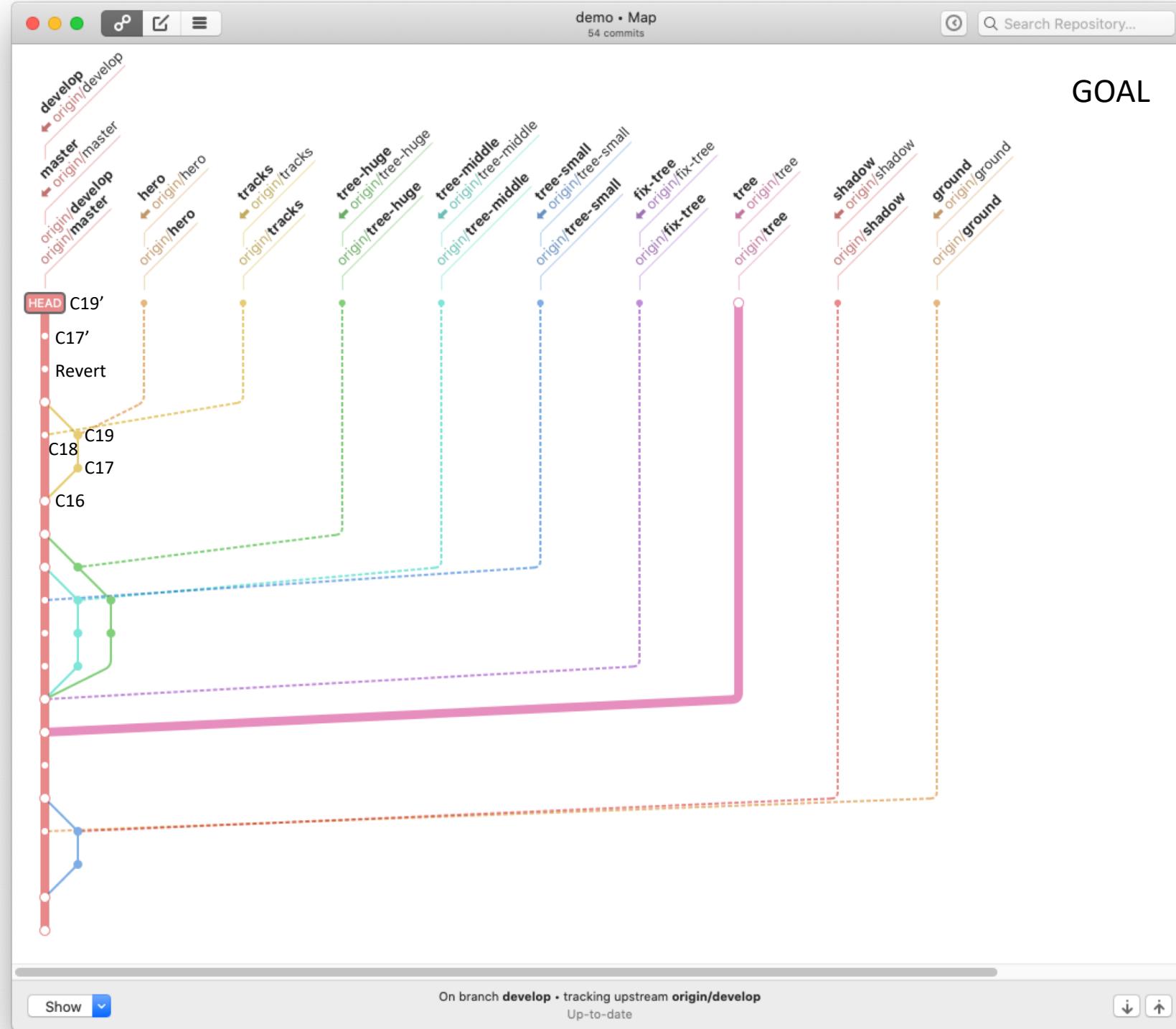
- Tại nhánh develop tạo nhánh tracks và push commit C16.
- Tại nhánh tracks tạo nhánh hero và push commit C17.
- Trở lại nhánh tracks và push commit C18, merge nhánh tracks vào nhánh develop.
- Trở lại nhánh hero và push commit C19, merge nhánh hero vào nhánh develop.
- Merge nhánh develop vào nhánh master.

Lúc này QC đã test nhánh develop OK, nhưng chưa cần tính năng tại nhánh hero mà yêu cầu deploy trước tính năng tracks.

Yêu cầu:

- Revert commit merge từ develop vào master, giữ lại code trên nhánh tracks.
- Lấy lại code của nhánh hero vào nhánh master.
- Chuyển nhánh develop về lại vị trí commit hiện tại của nhánh master

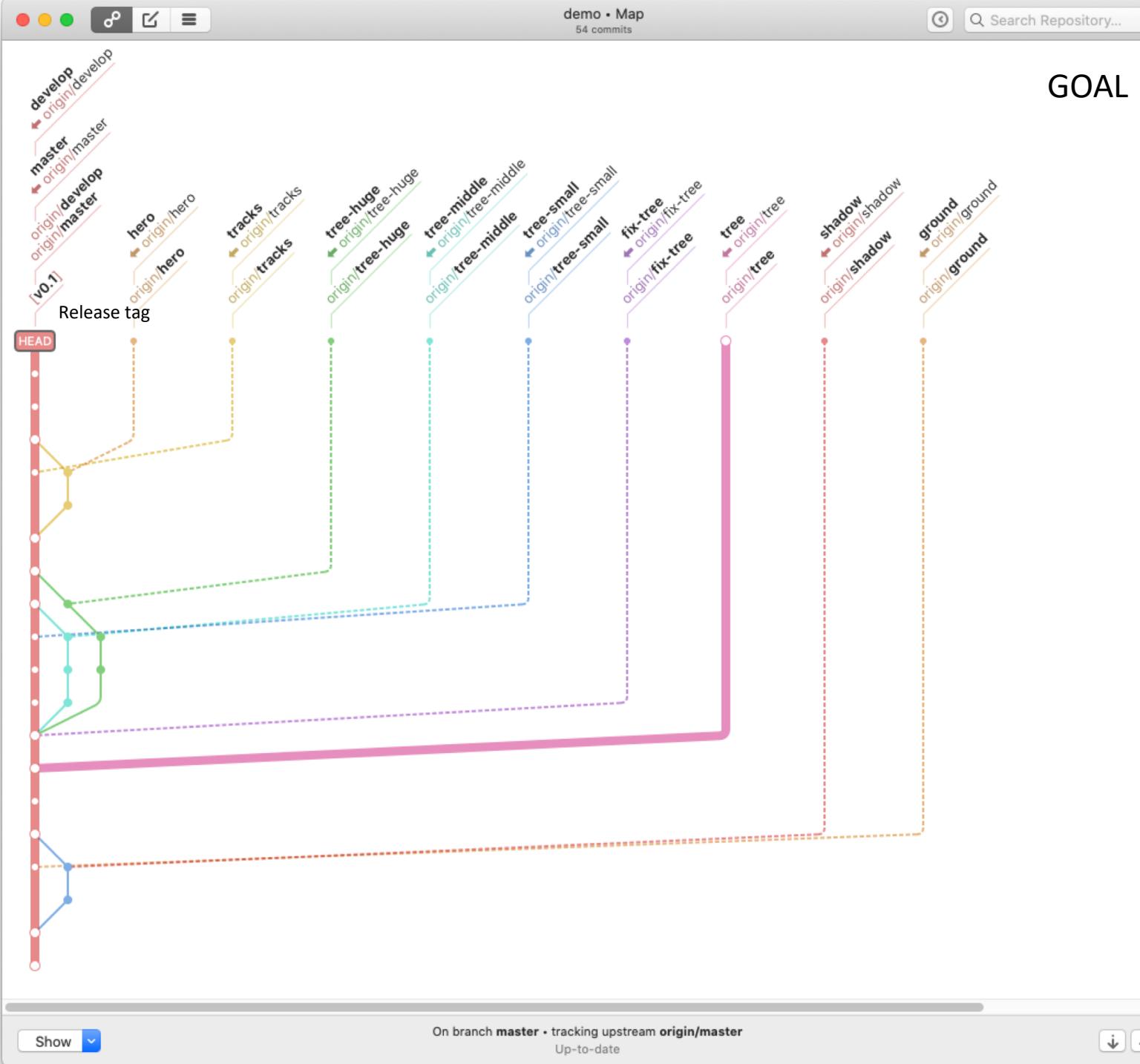




BT 5:

Yêu cầu:

- Tạo 1 tag trên nhánh master với tên v0.1.
- Sử dụng lệnh git archive để tải toàn bộ code của version v0.1 vào thư mục Document để lưu trữ.
- Mở folder đã tải, chạy file index.html để xem thành quả của mình.



Good luck