Homework #1 – Git Practice

Git Tutorial

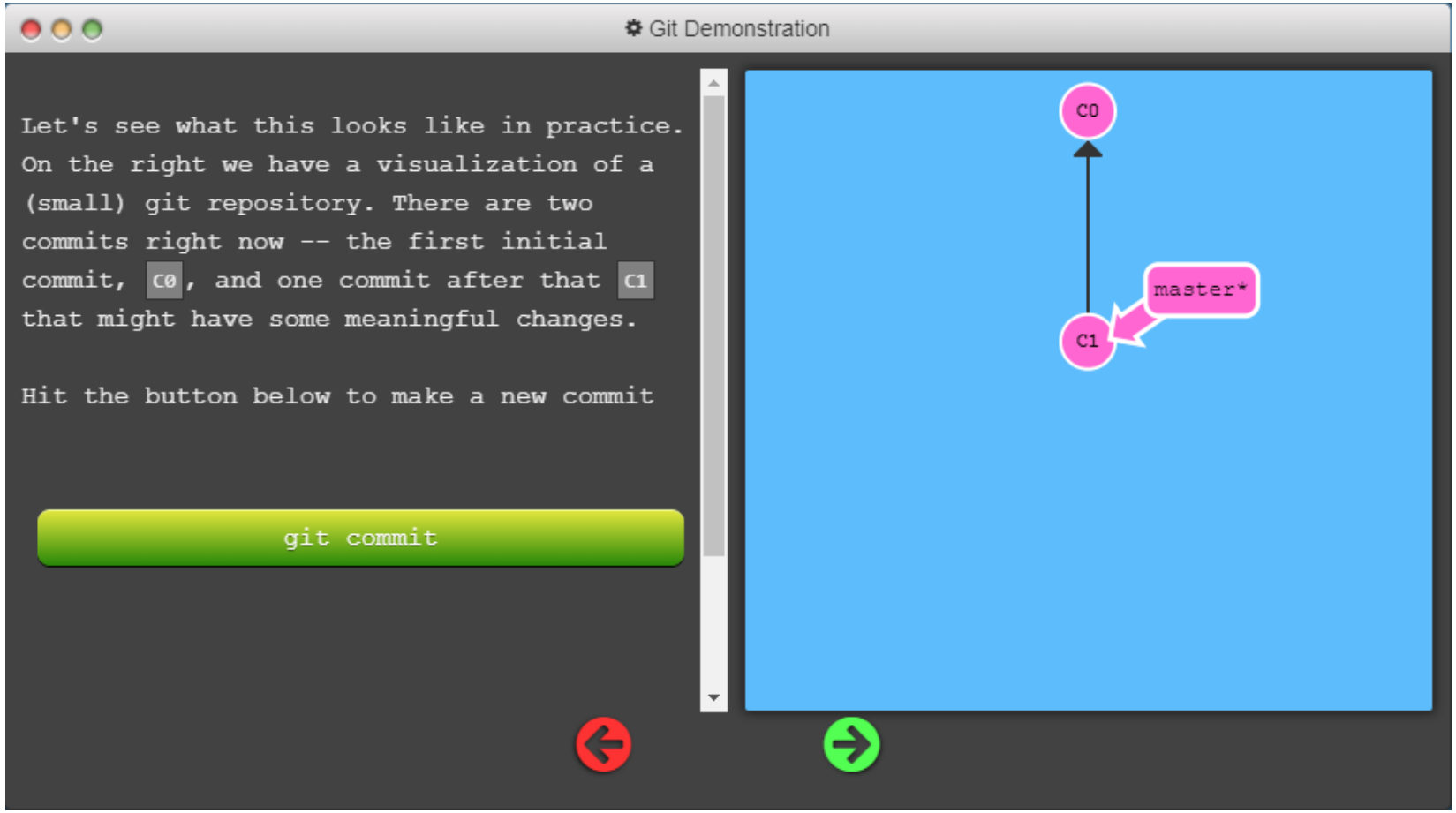
- https://try.github.io/
  - Learn by doing > Learn Git Branching > 1
- https://learngitbranching.js.org/

Homework #1 – Git Practice

- Introduction to Git Commits
  - Repository*:* 하나의 프로젝트를 관리하기 위한 저장소
    - 프로젝트에 필요한 다양한 형식의 데이터를 저장할 수 있음
      - 소스코드
      - 폴더, 파일, 이미지, 비디오, 스프레드 시트 등
      - README, License 파일
  - *Commit:* 저장소의 스냅샷을 만들고 저장하는 것
    - 이 방법으로 수정사항에 대한 것들을 추적할 수 있음

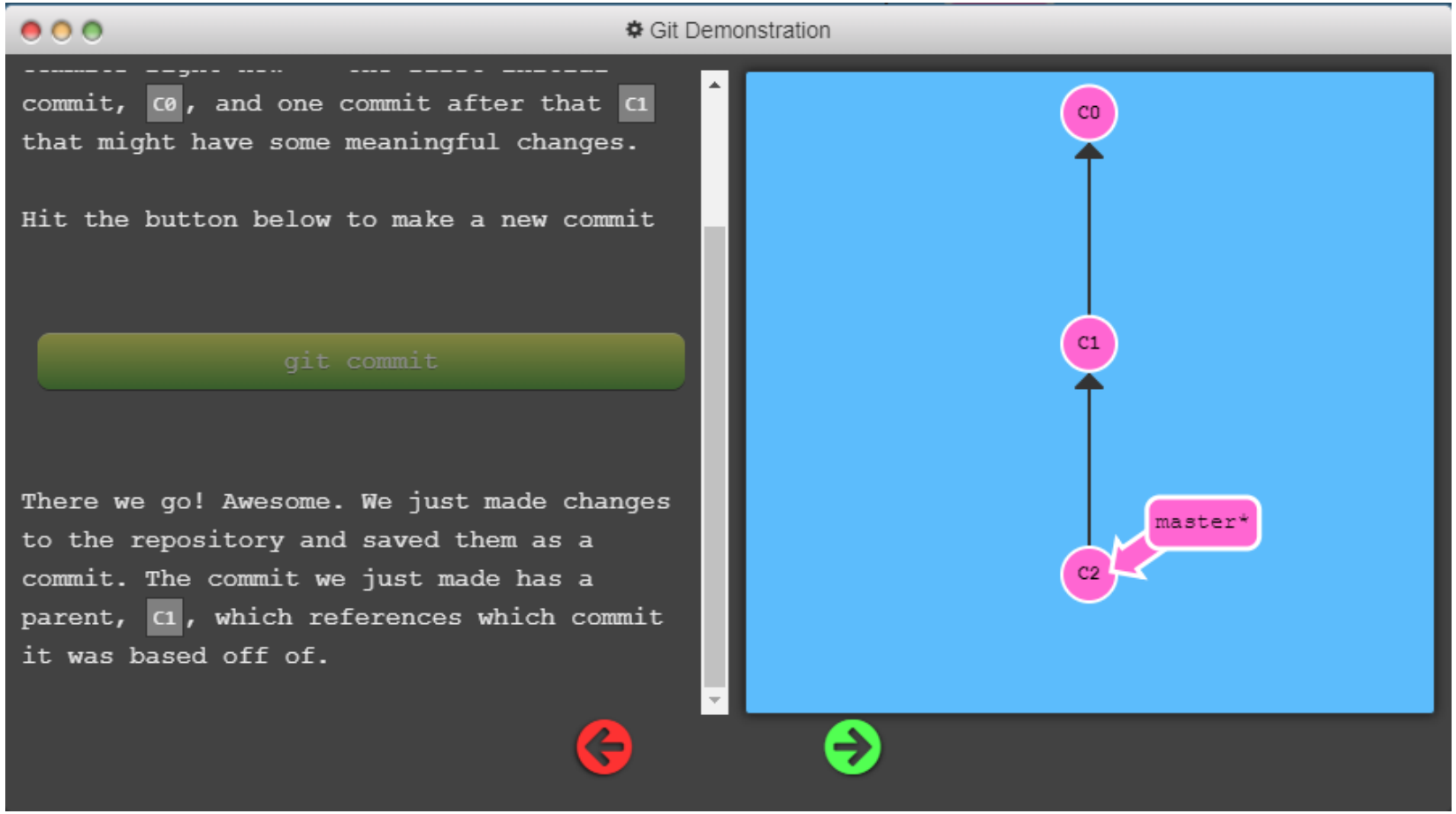
Homework #1 – Git Practice

$ git commit



Homework #1 – Git Practice

$ git commit

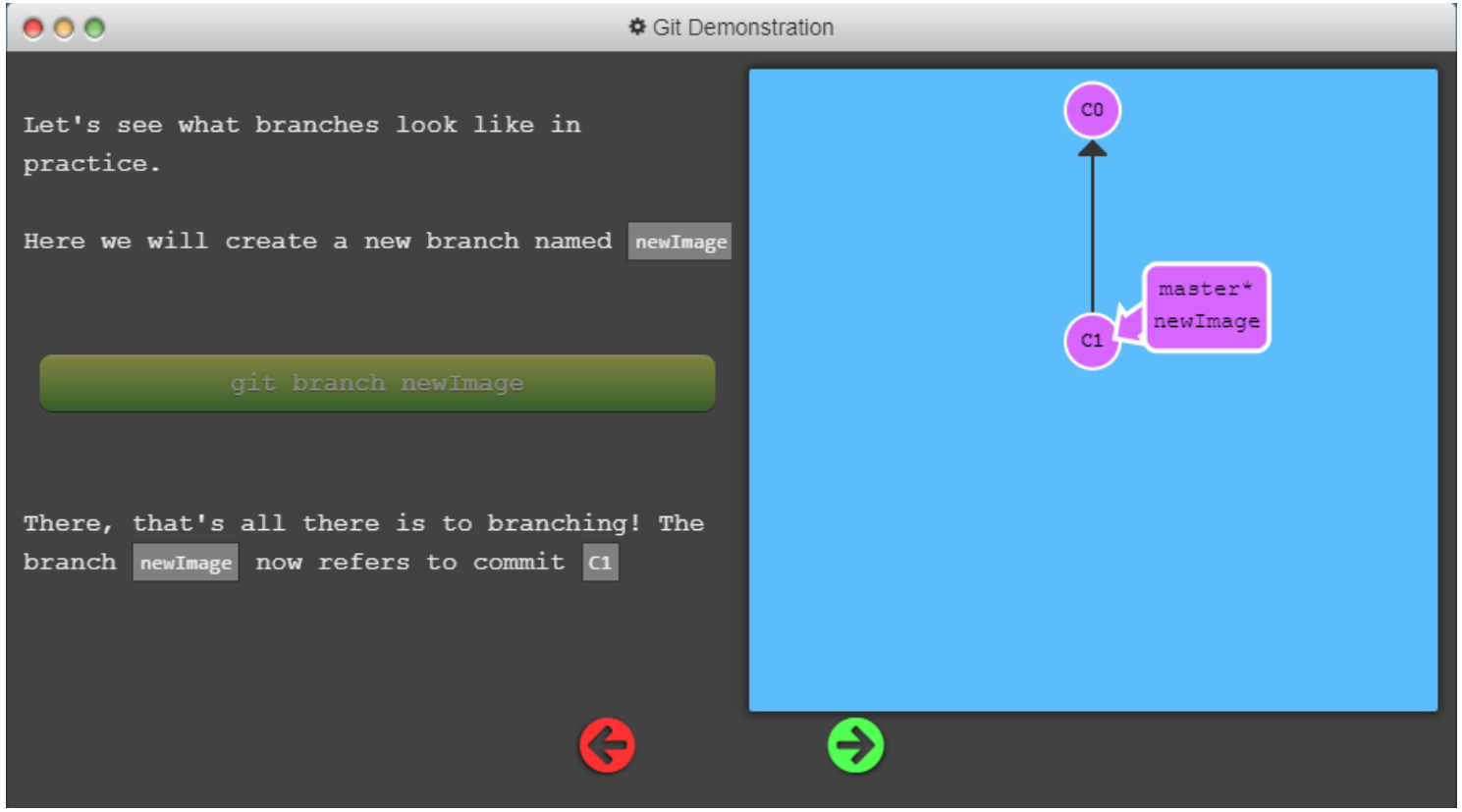


Homework #1 – Git Practice

- Branching in Git
  - Branch: A pointer that indicates a specific commit
  - Great for making changes:
    - Add a new function, testing, finding a bug, concurrent work
  - “branch early, and branch often”

Homework #1 – Git Practice

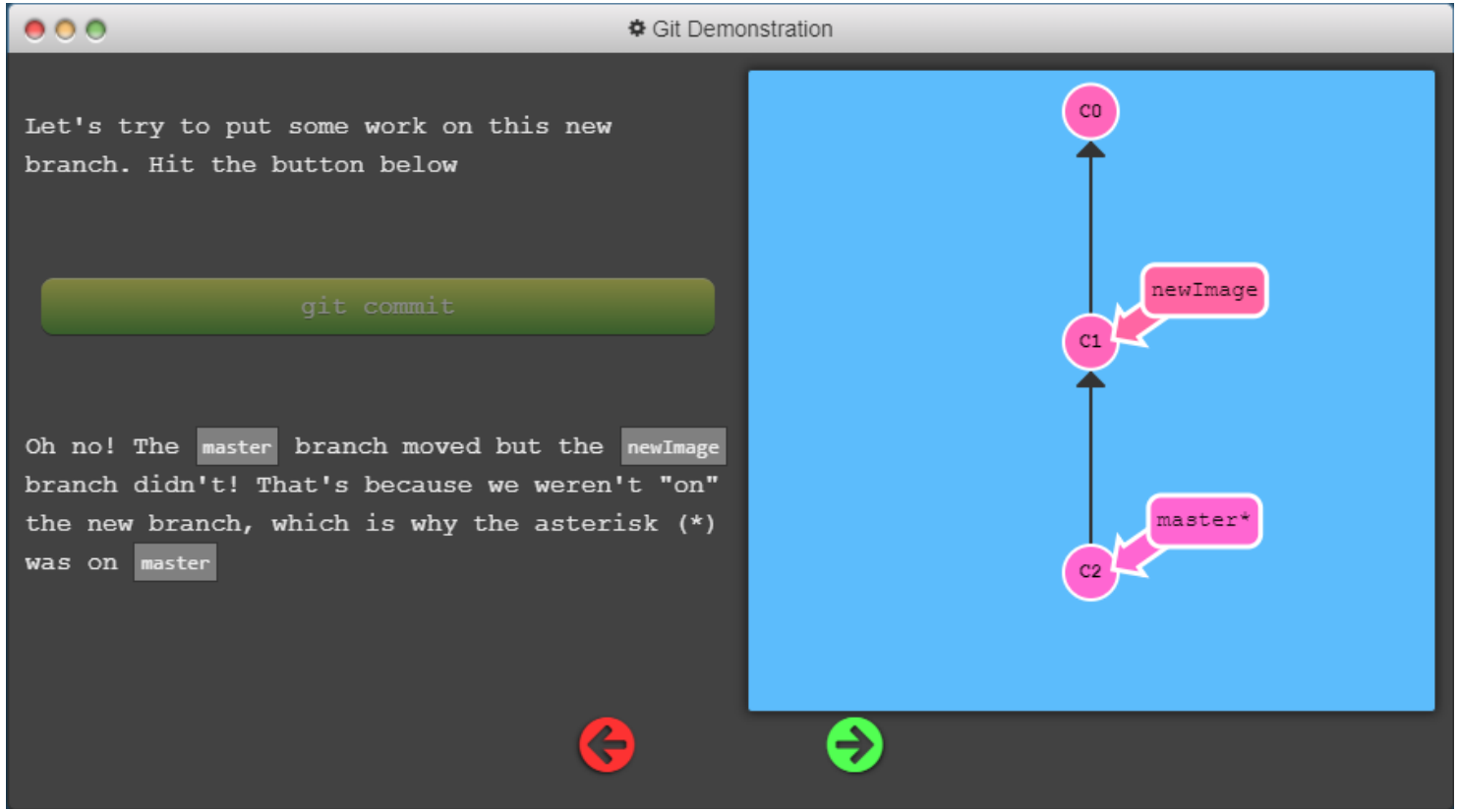
$ git checkout -b [new_branch_name]



Create a new branch

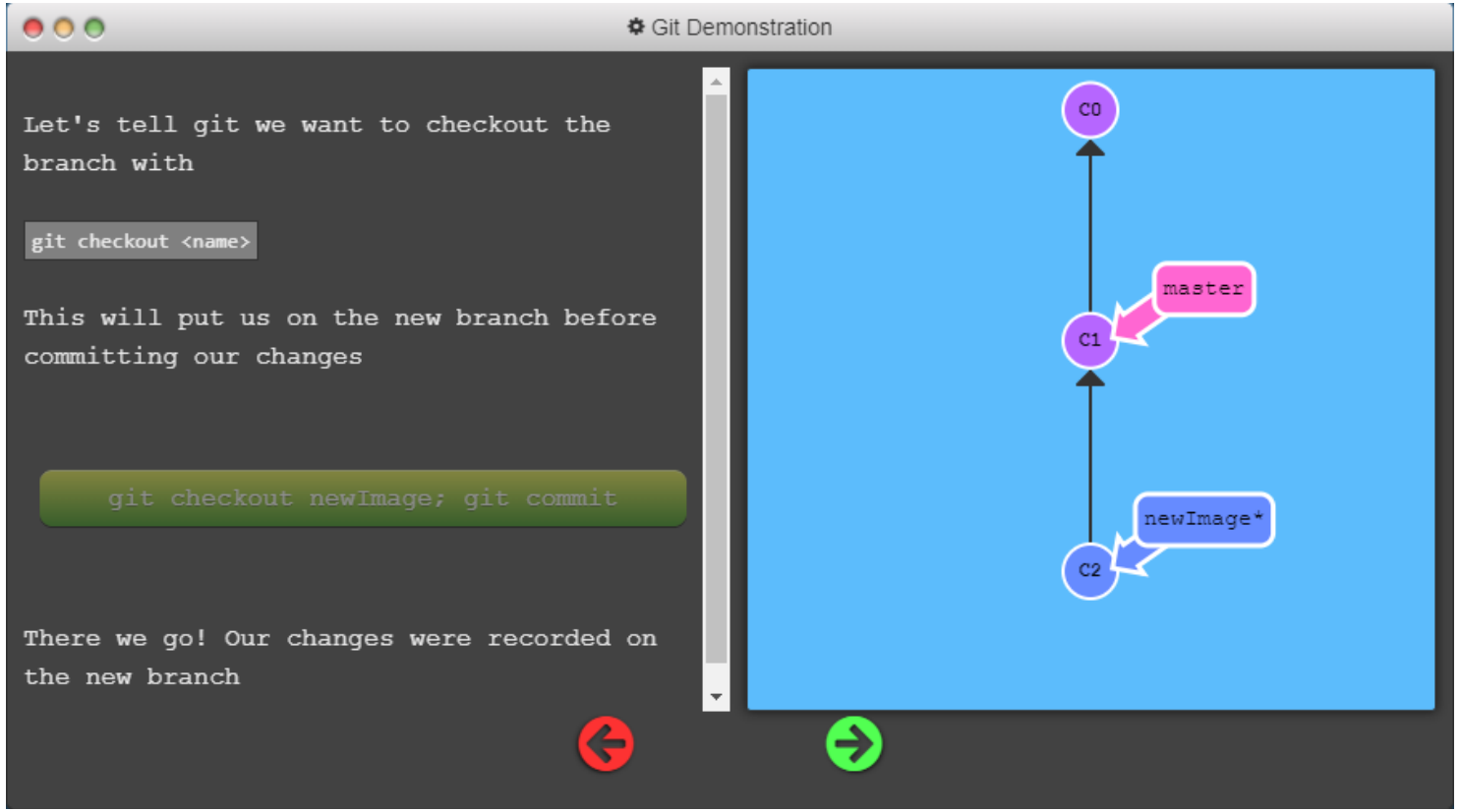
Homework #1 – Git Practice

$ git commit



Homework #1 – Git Practice

$ git checkout [existing_branch_name]

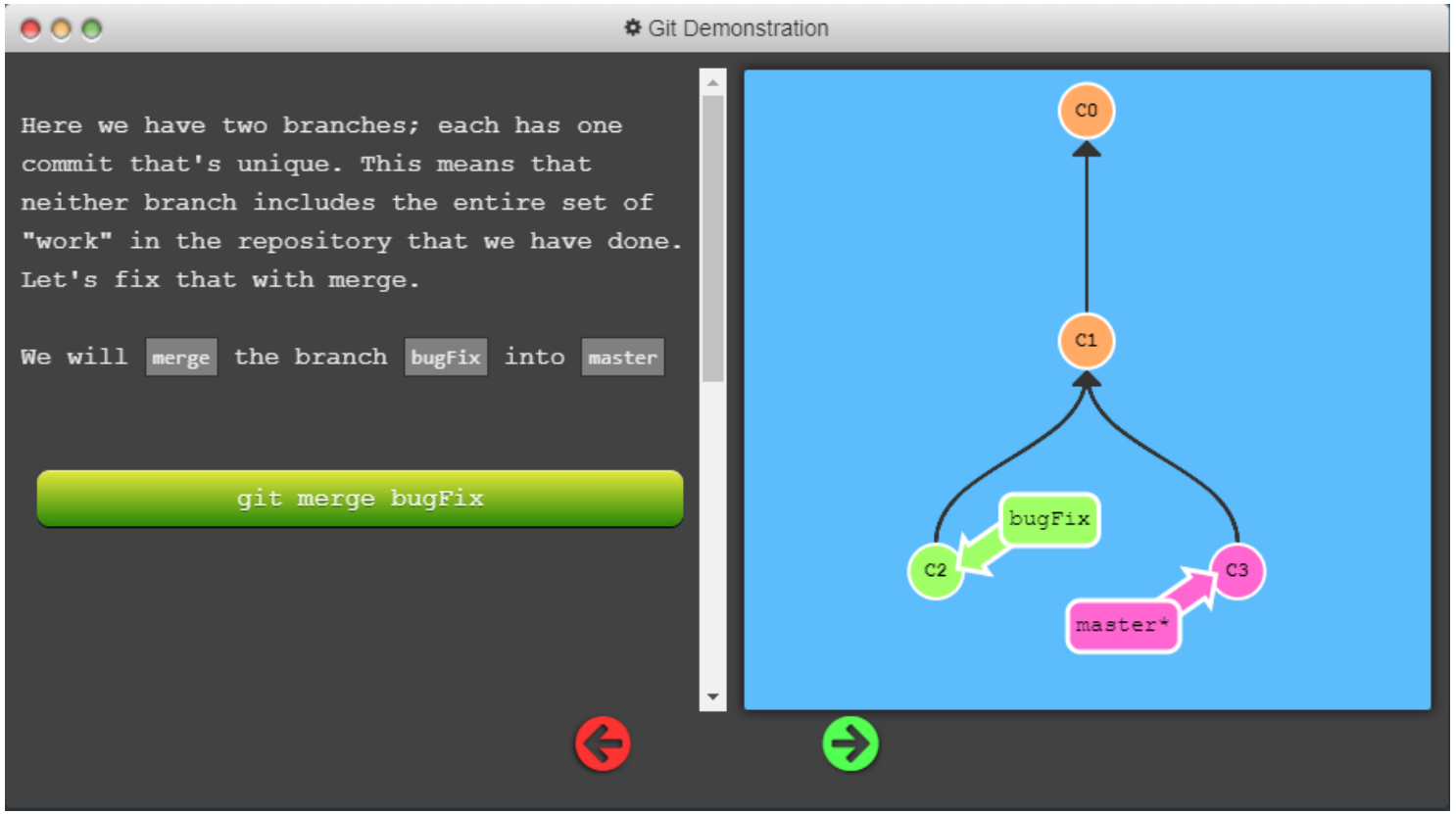


Switch to another branch

Homework #1 – Git Practice

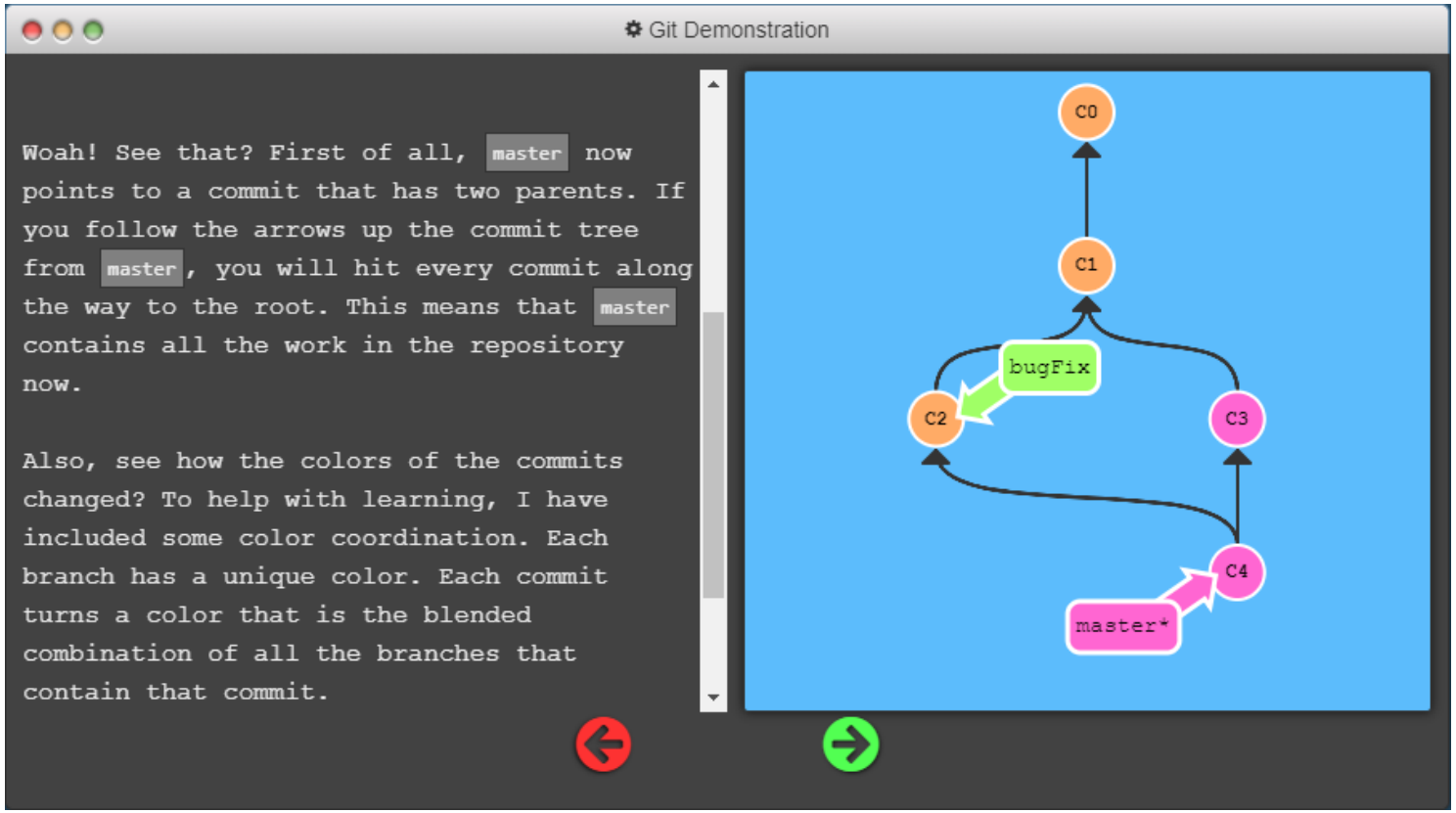
- Merging in Git
  - *Merge*: Merging two different branches

Homework #1 – Git Practice



Homework #1 – Git Practice

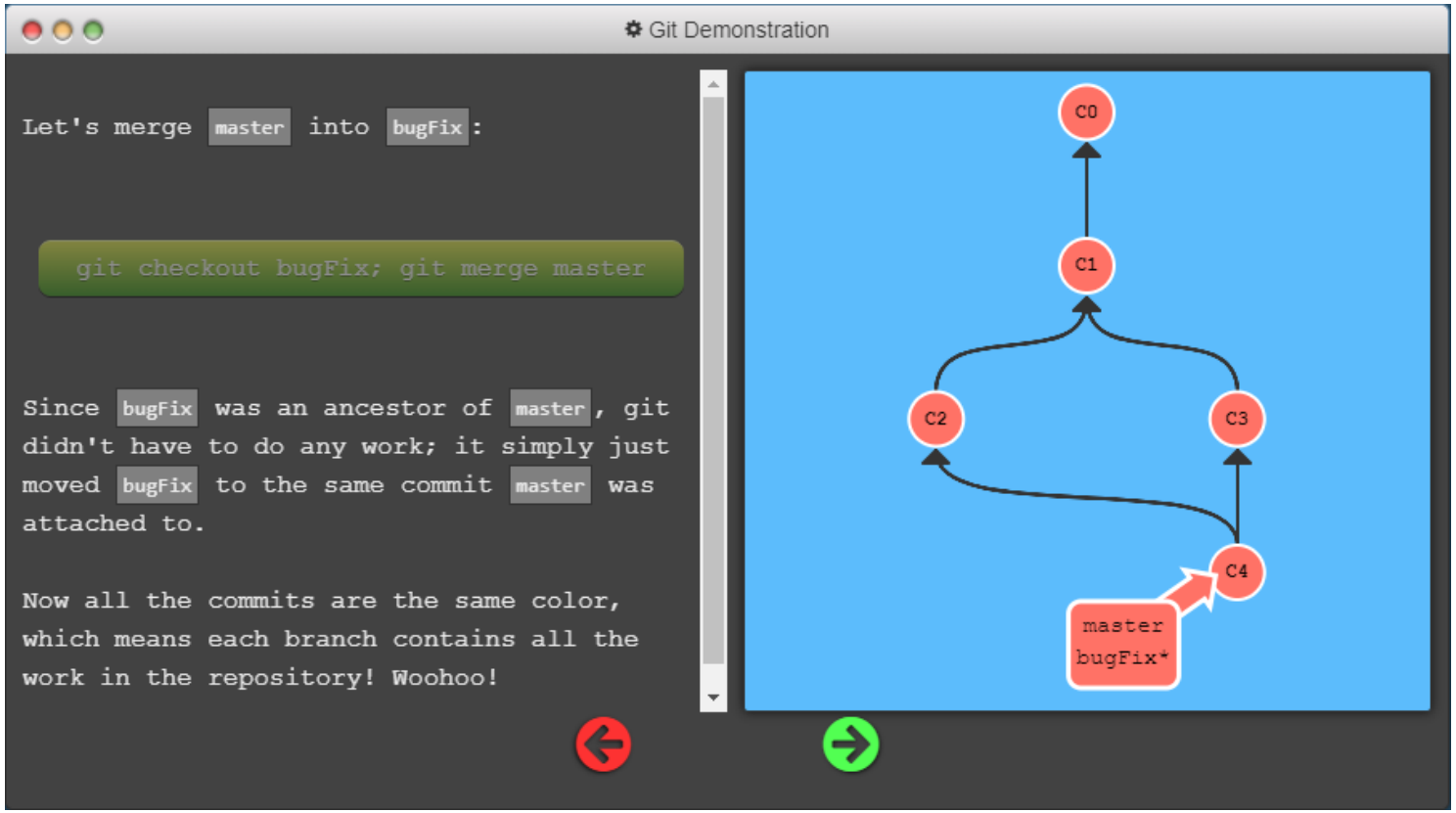
$ git merge bugFix



Merge bugFix branch with master branch

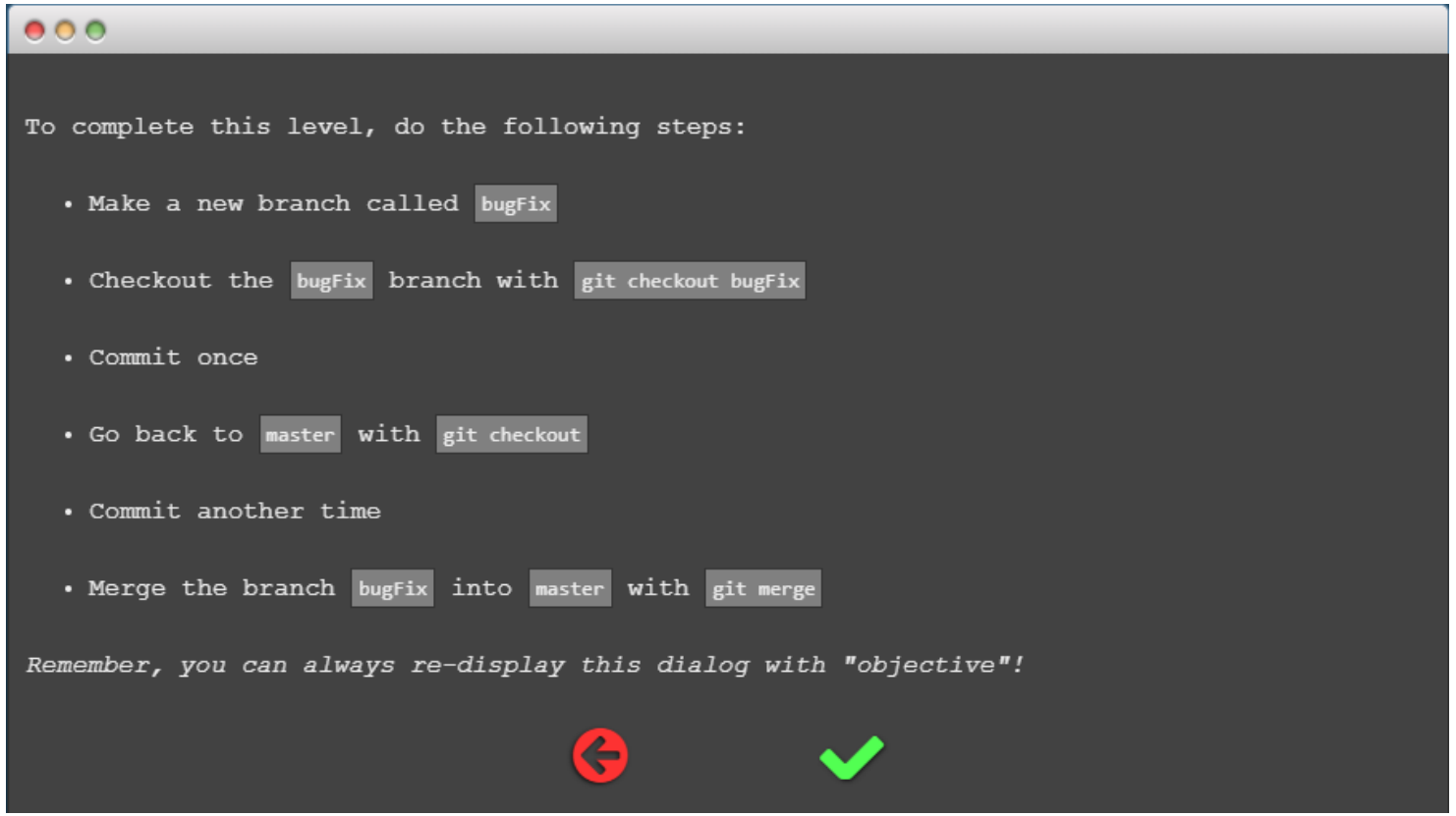
Homework #1 – Git Practice

$ git checkout bugfix; git merge master



Switch to another branch and then merge master branch with bugFix branch

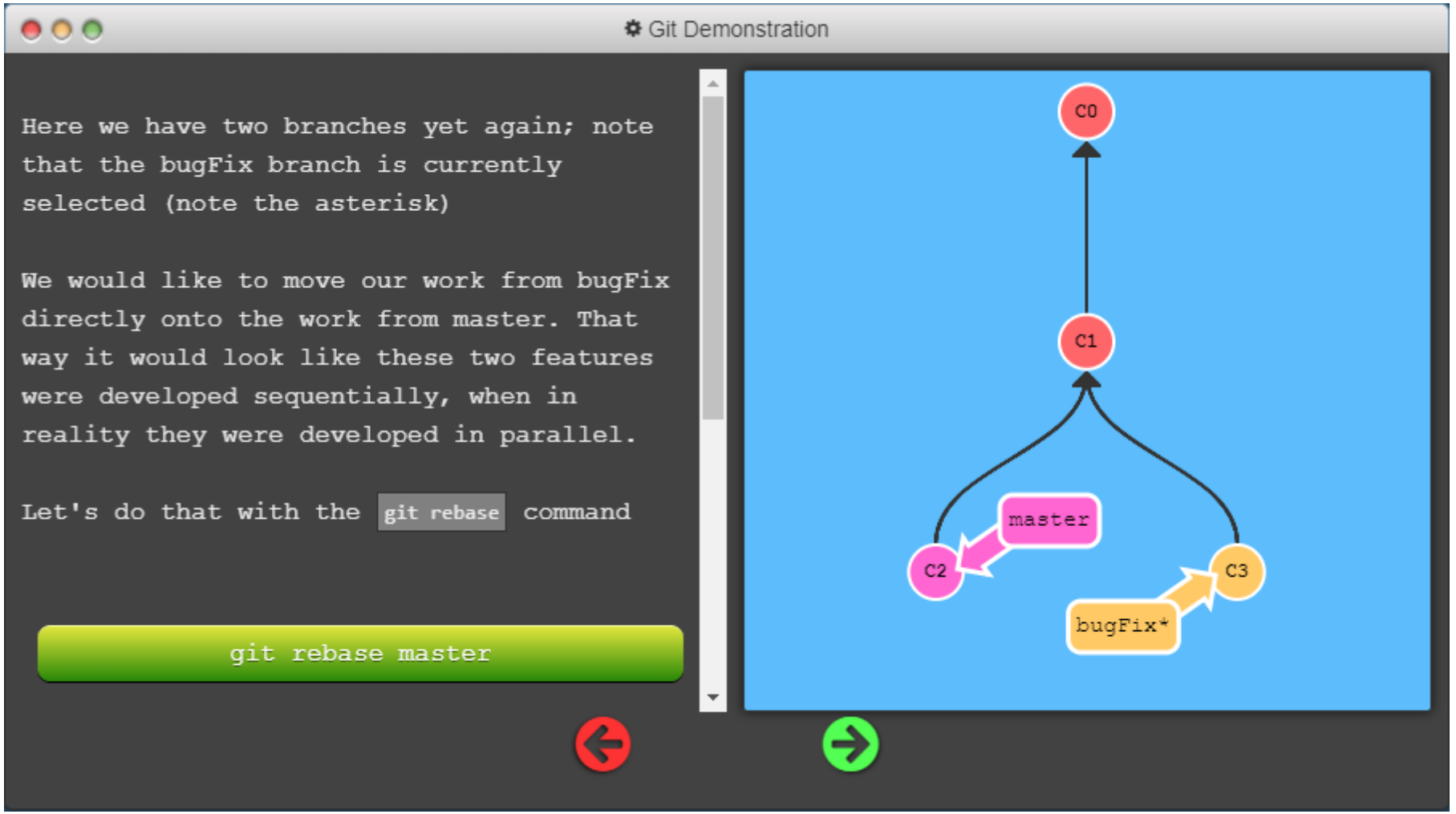
Homework #1 – Git Practice



Homework #1 – Git Practice

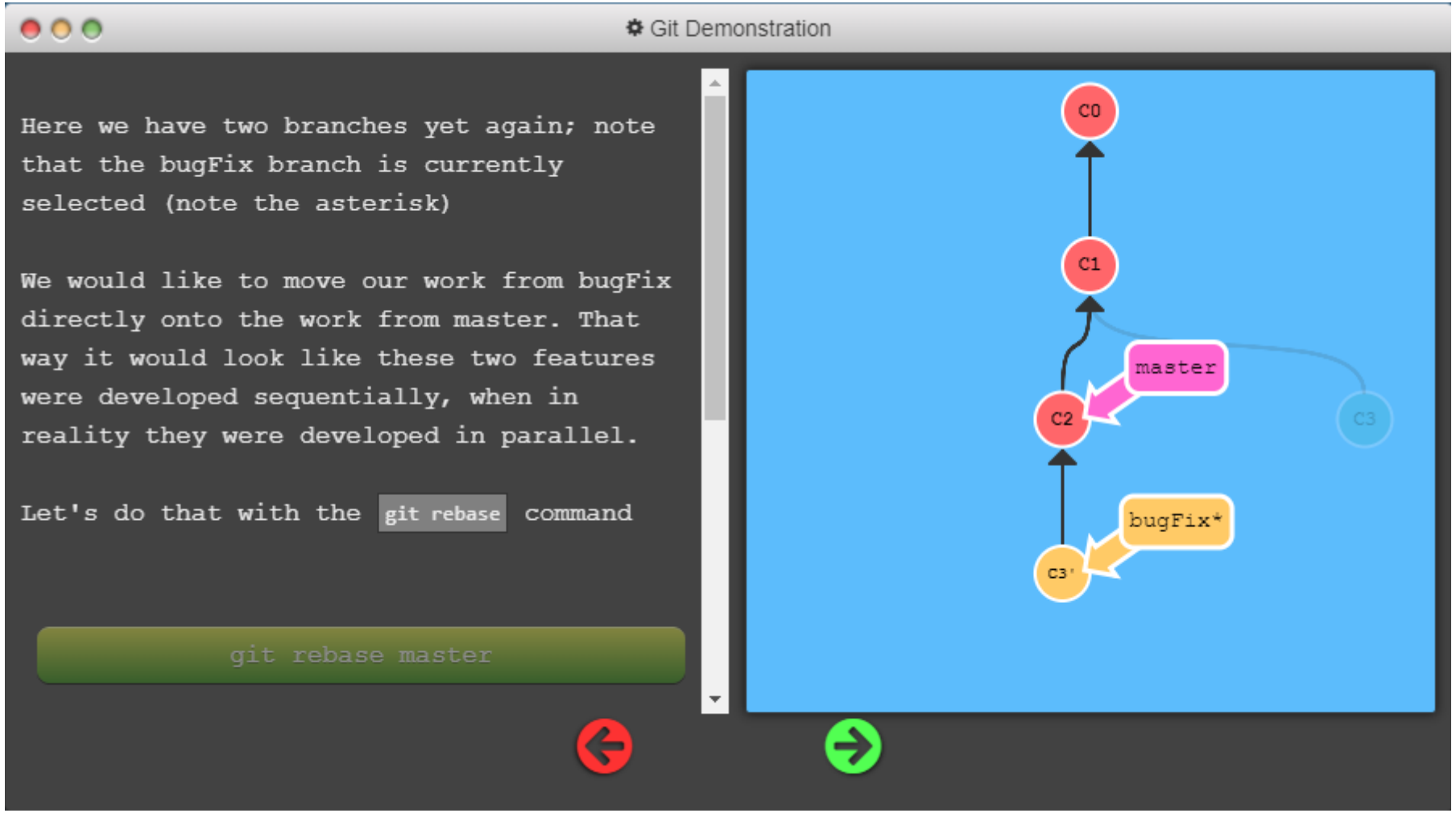
- Rebase Introduction
  - Another way to merge branches
  - Easy to manage commits

Homework #1 – Git Practice



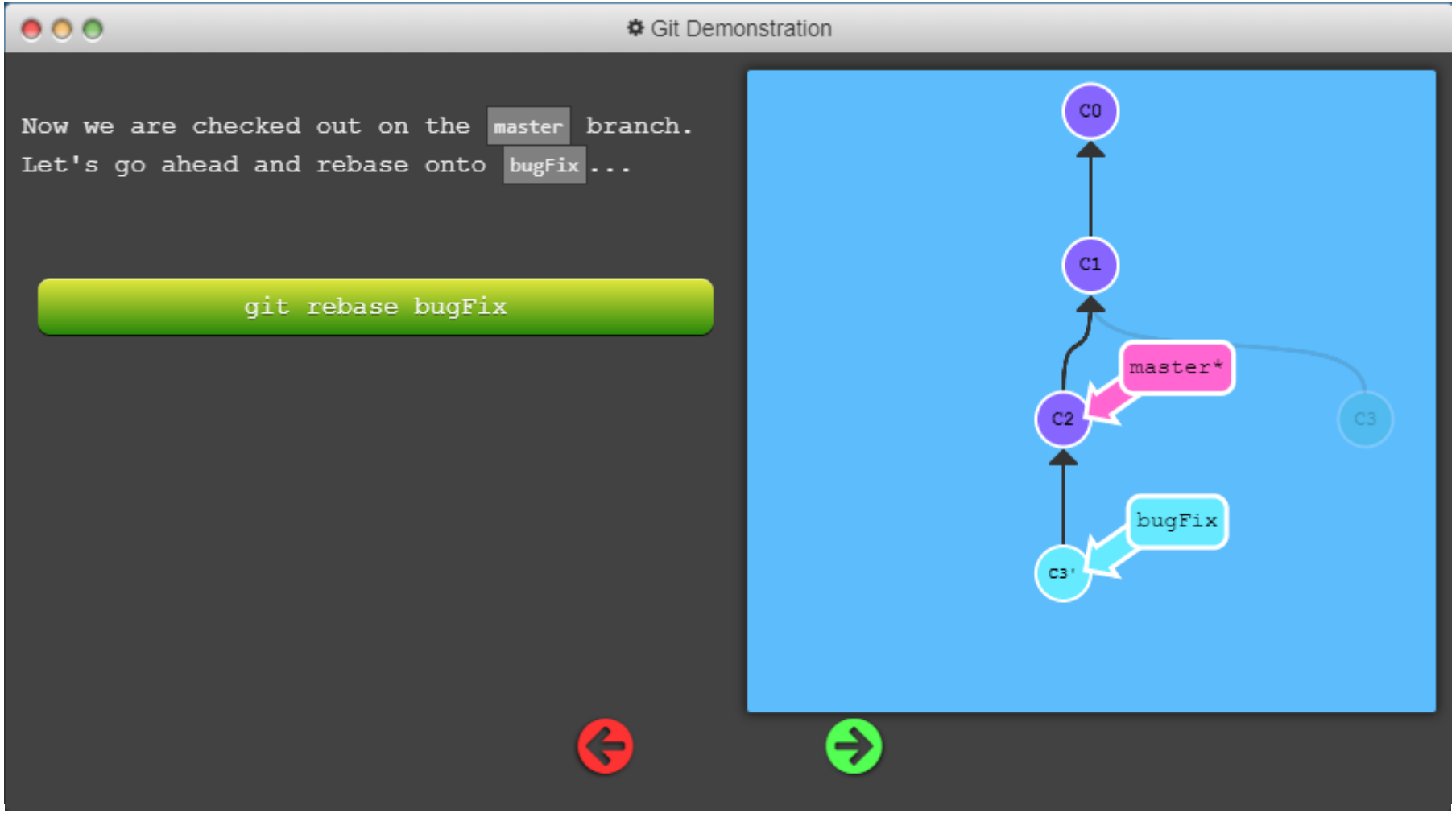
Homework #1 – Git Practice

$ git rebase master



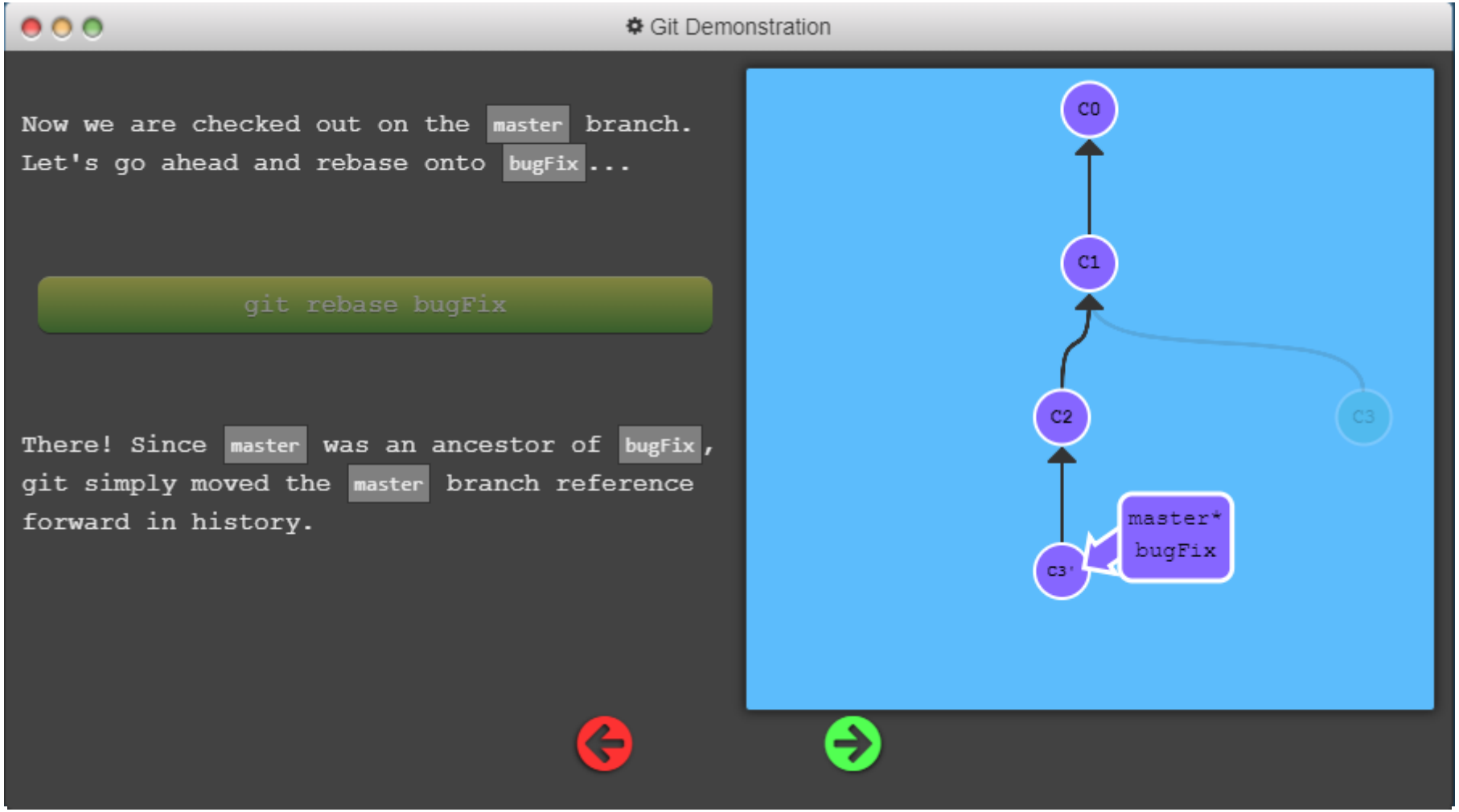
Homework #1 – Git Practice

$ git checkout master

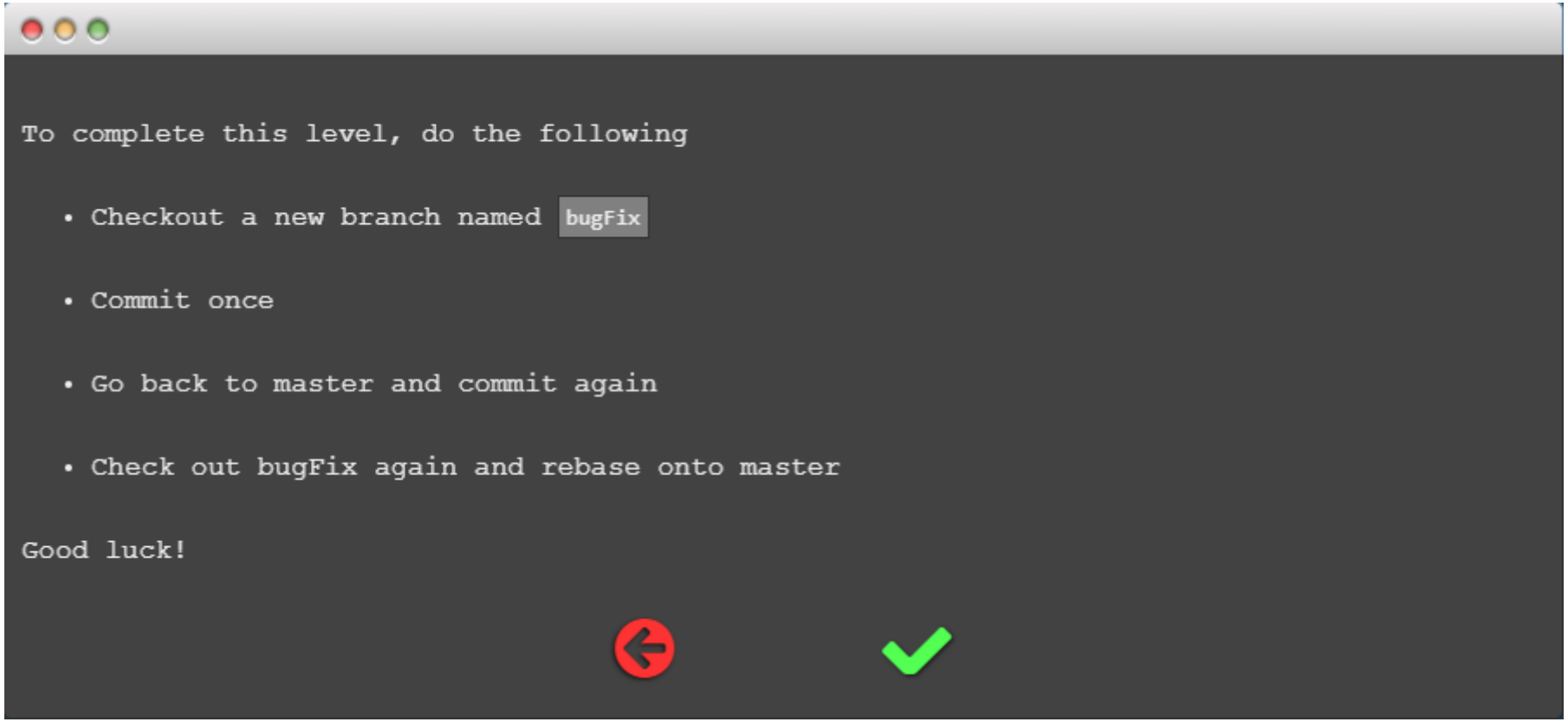


Homework #1 – Git Practice

$ git rebase bugFix



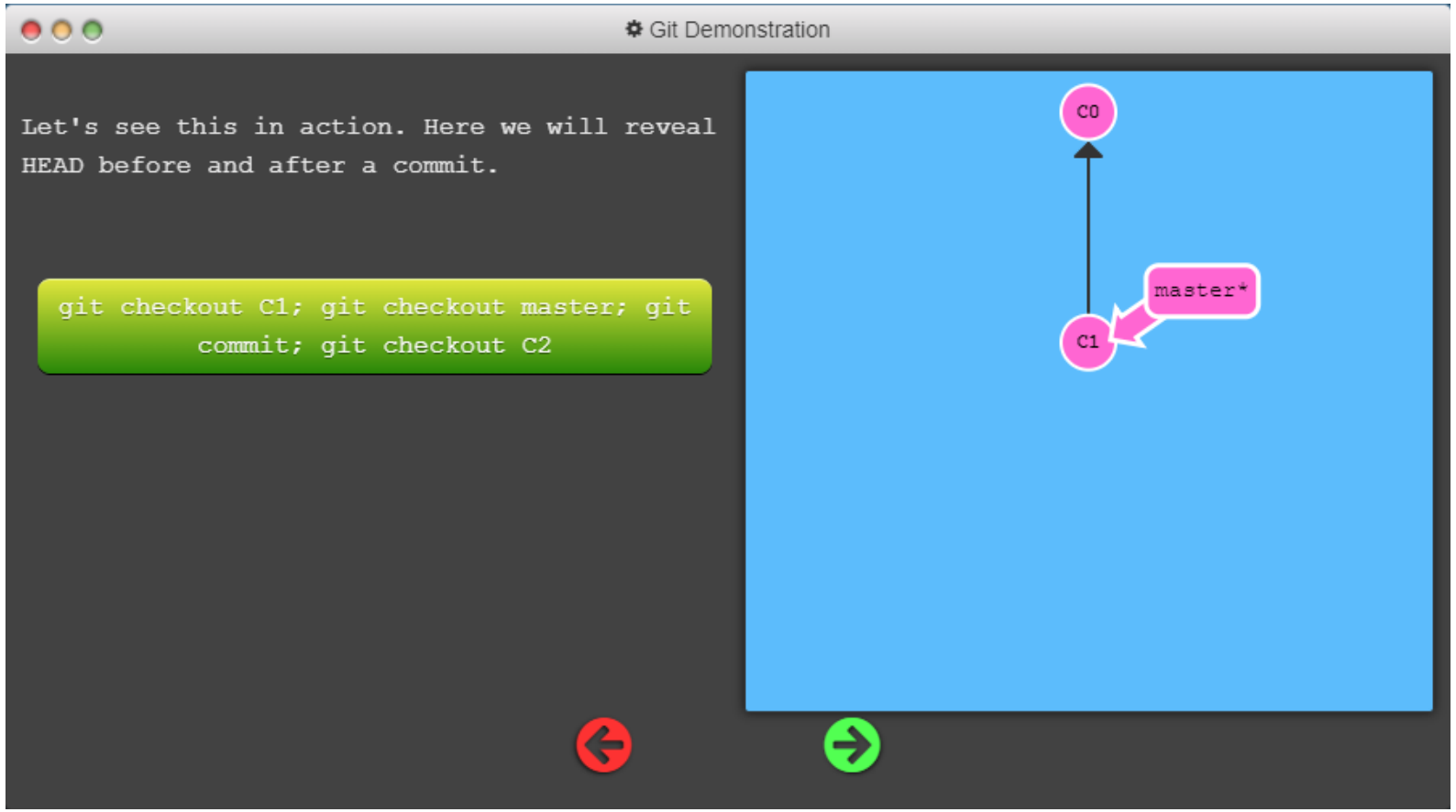
Homework #1 – Git Practice



Homework #1 – Git Practice

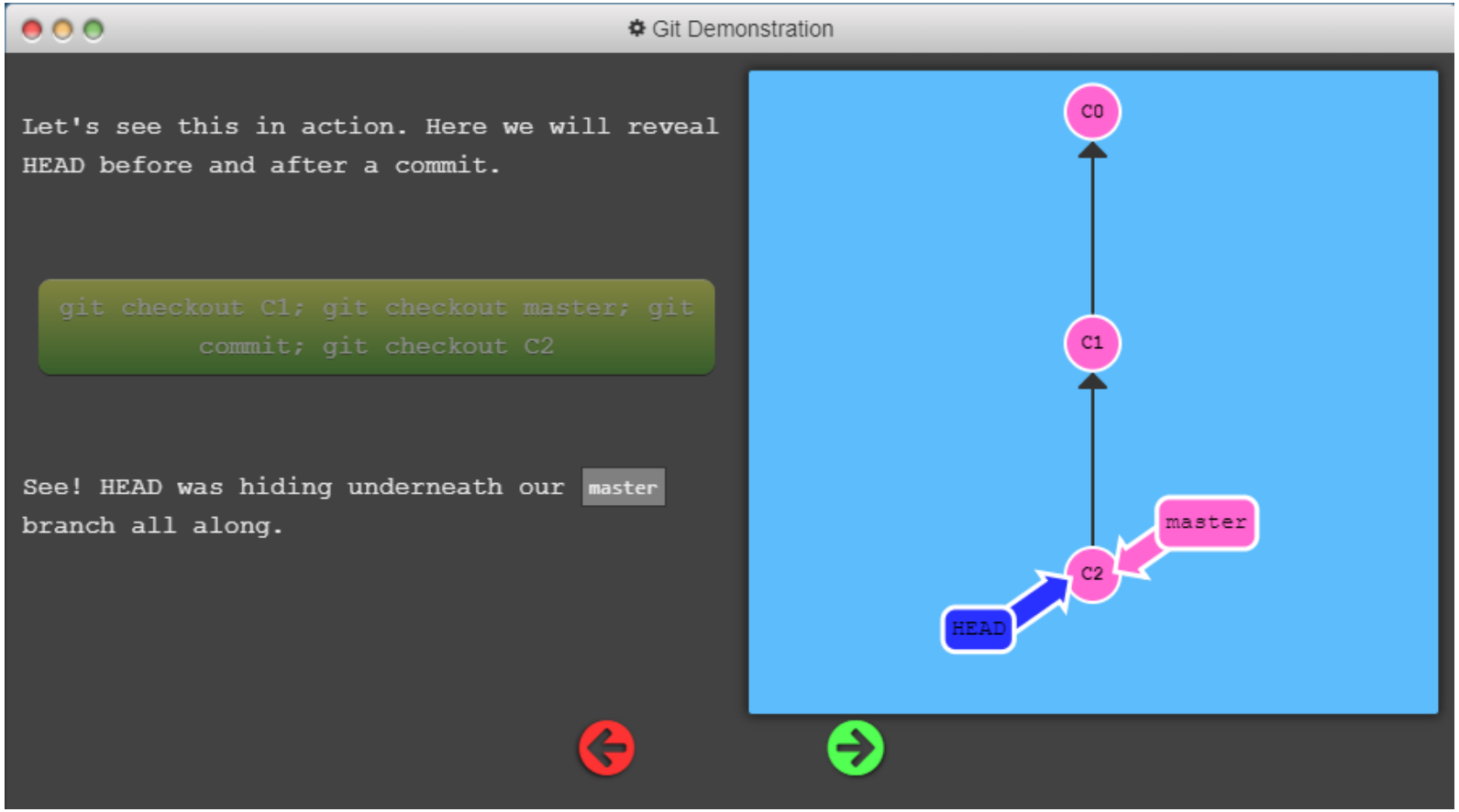
- Detach yo' HEAD
  - HEAD: A pointer that indicates the currently checked out commit
    - The working commit
    - The most recent commit

Homework #1 – Git Practice

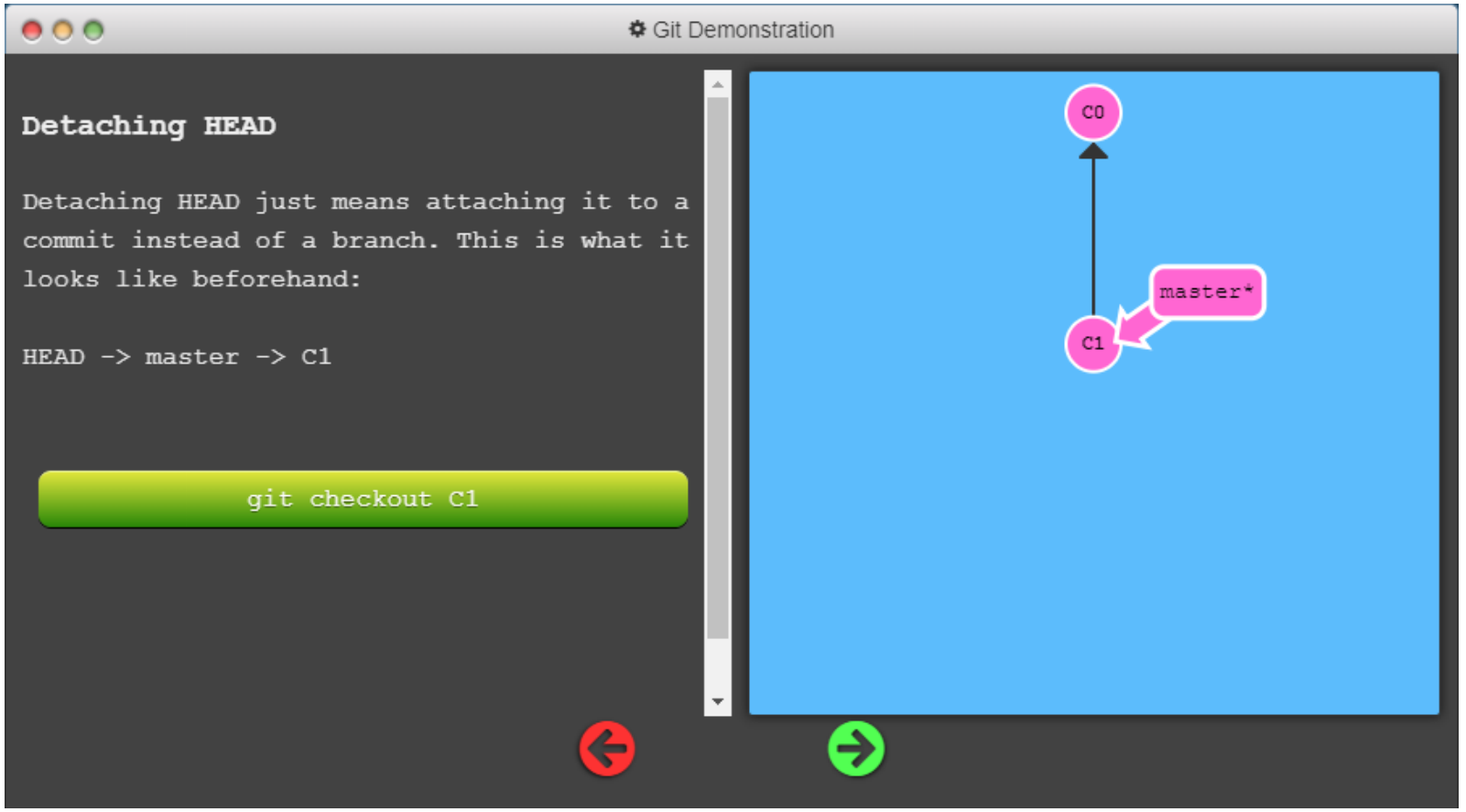


Homework #1 – Git Practice

$ git checkout C1; git checkout master; git commit; git checkout C2

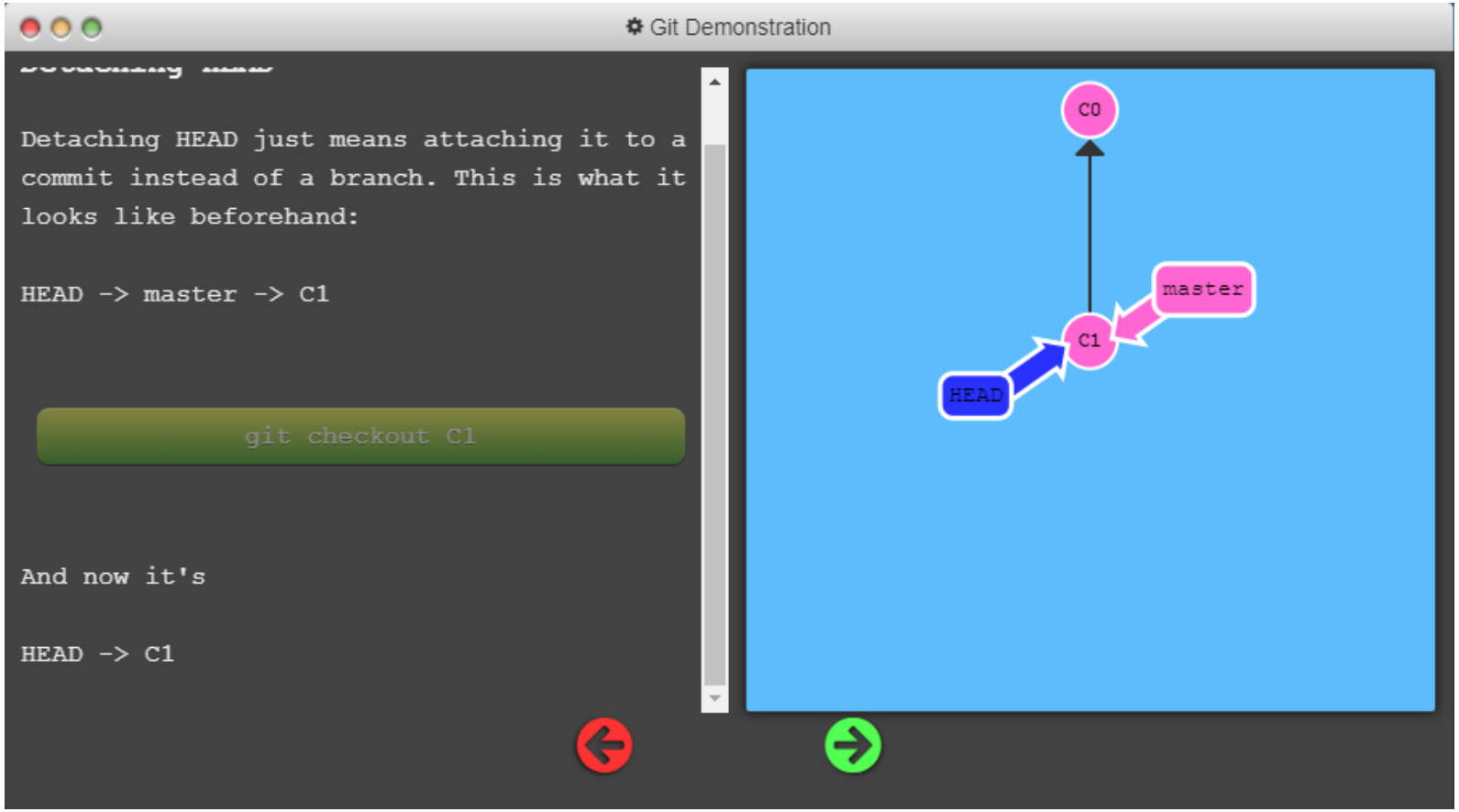


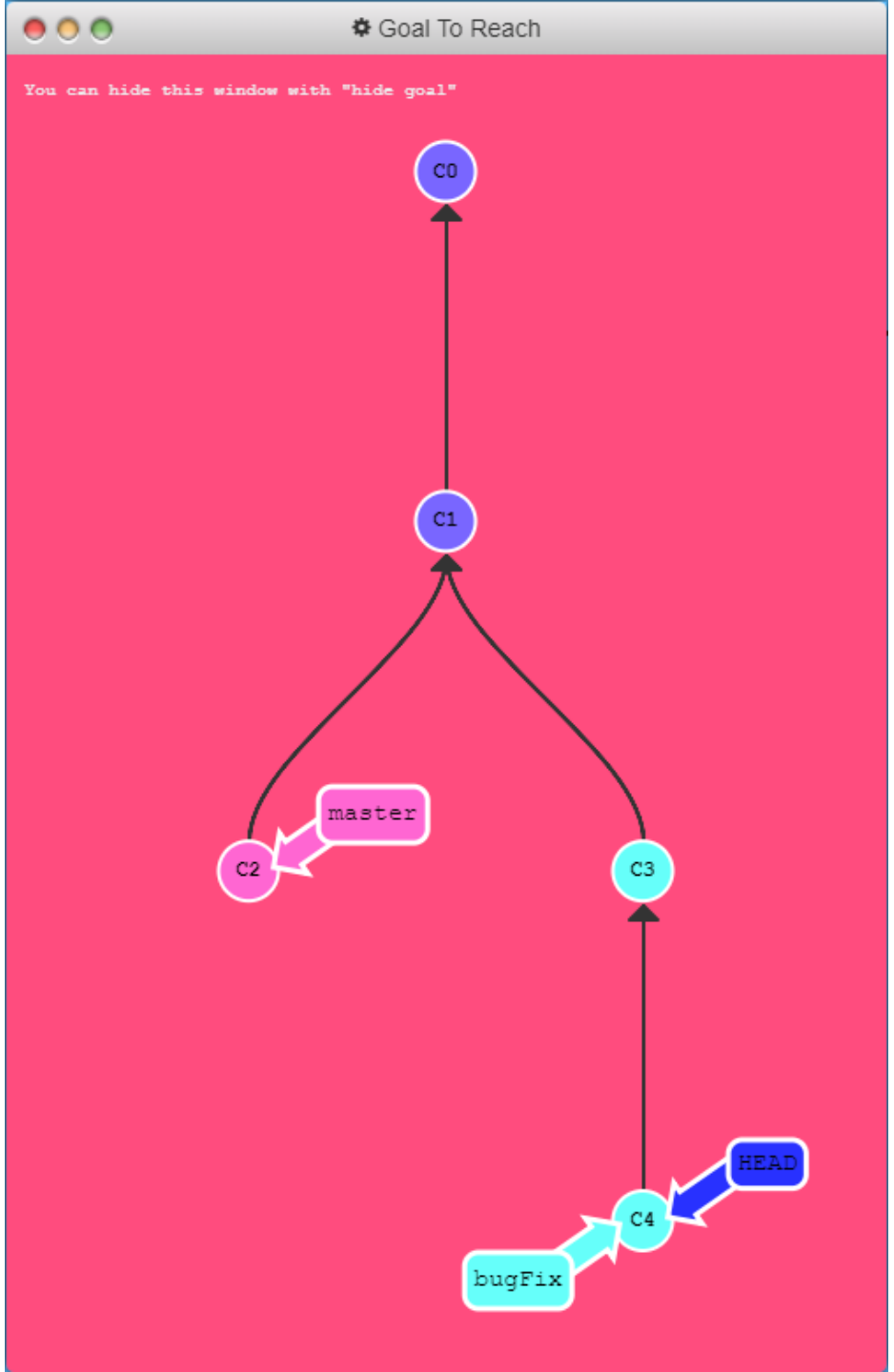
Homework #1 – Git Practice



Homework #1 – Git Practice

$ git checkout C1



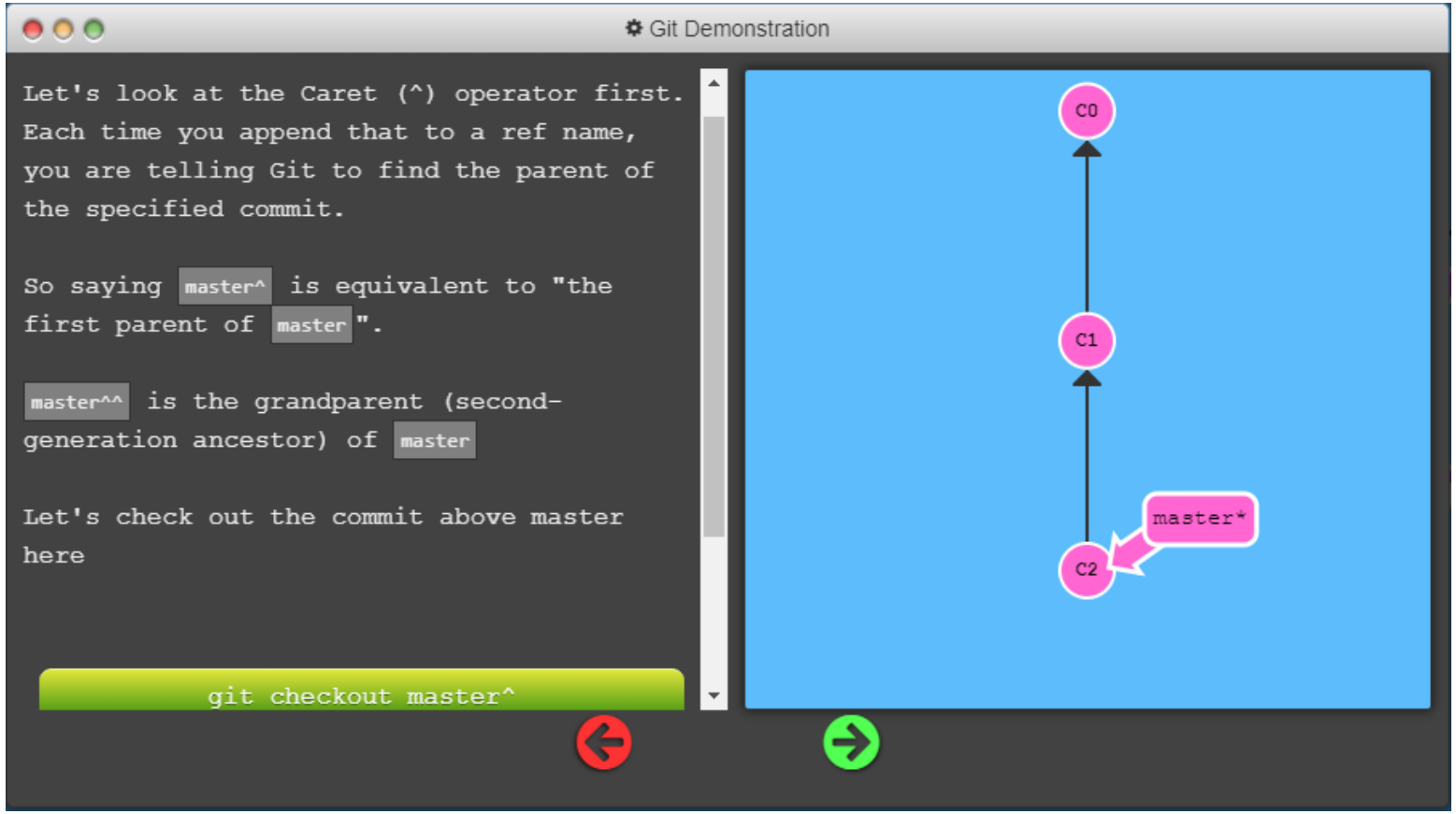

Goal To Reach
You can hide this window with "hide goal"
C0
C1
master
C2
C3
HEAD
C4
bugFix

Homework #1 – Git Practice

- Relative Refs
  - You can move between commits using hash (absolute refs)
    - But cumbersome
  - It's easier with relative refs
    - ^: Move to the last commit (the one above)
    - ~<num>: move to the <num> of commits above

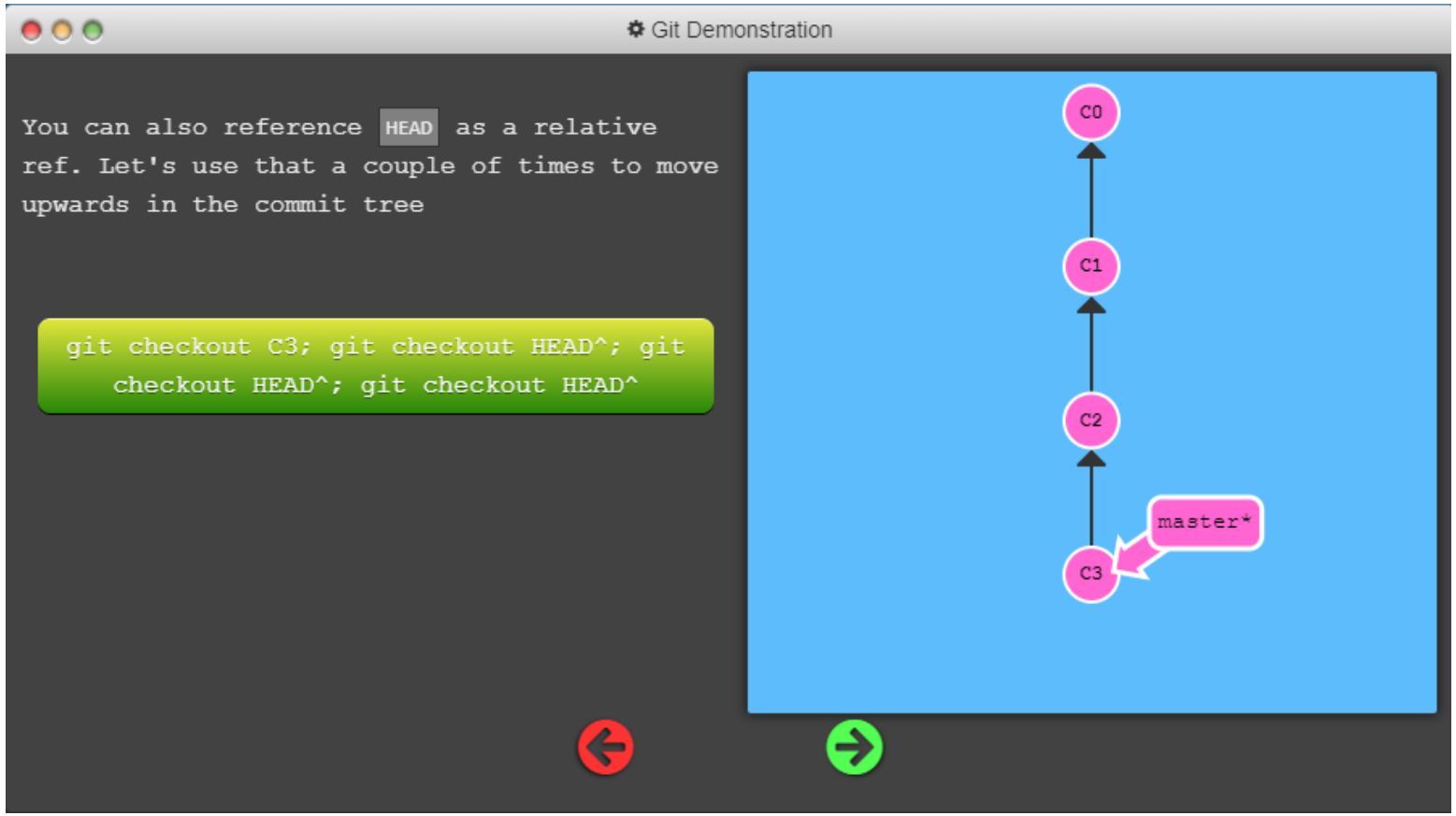
Homework #1 – Git Practice

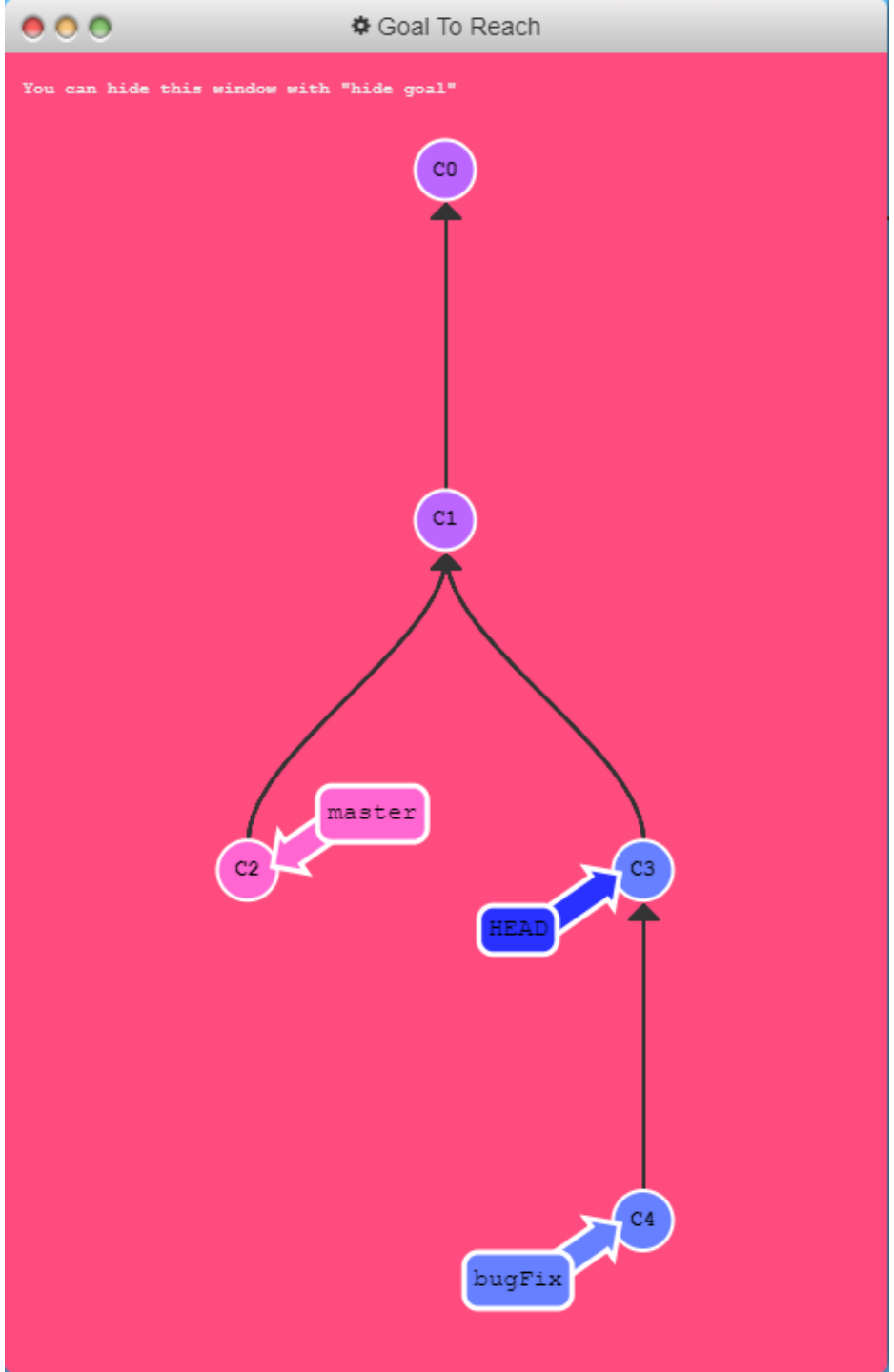
$ git checkout master^



Homework #1 – Git Practice

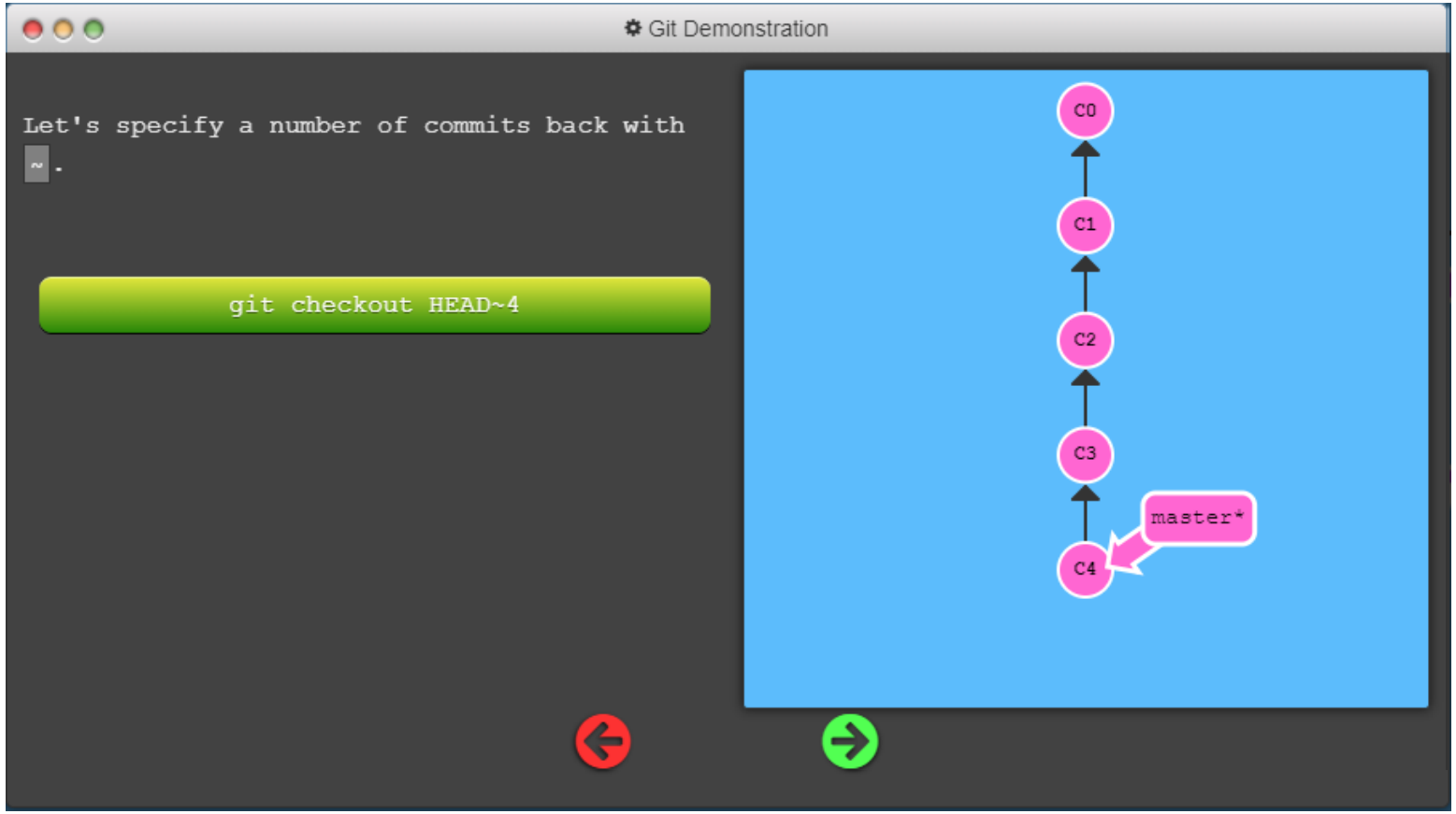
$ git checkout C3; git checkout HEAD^; git checkout HEAD^; git checkout HEAD^




Goal To Reach
You can hide this window with "hide goal"
C0
C1
master
C2
C3
HEAD
C4
bugFix

Homework #1 – Git Practice

$ git checkout HEAD~4

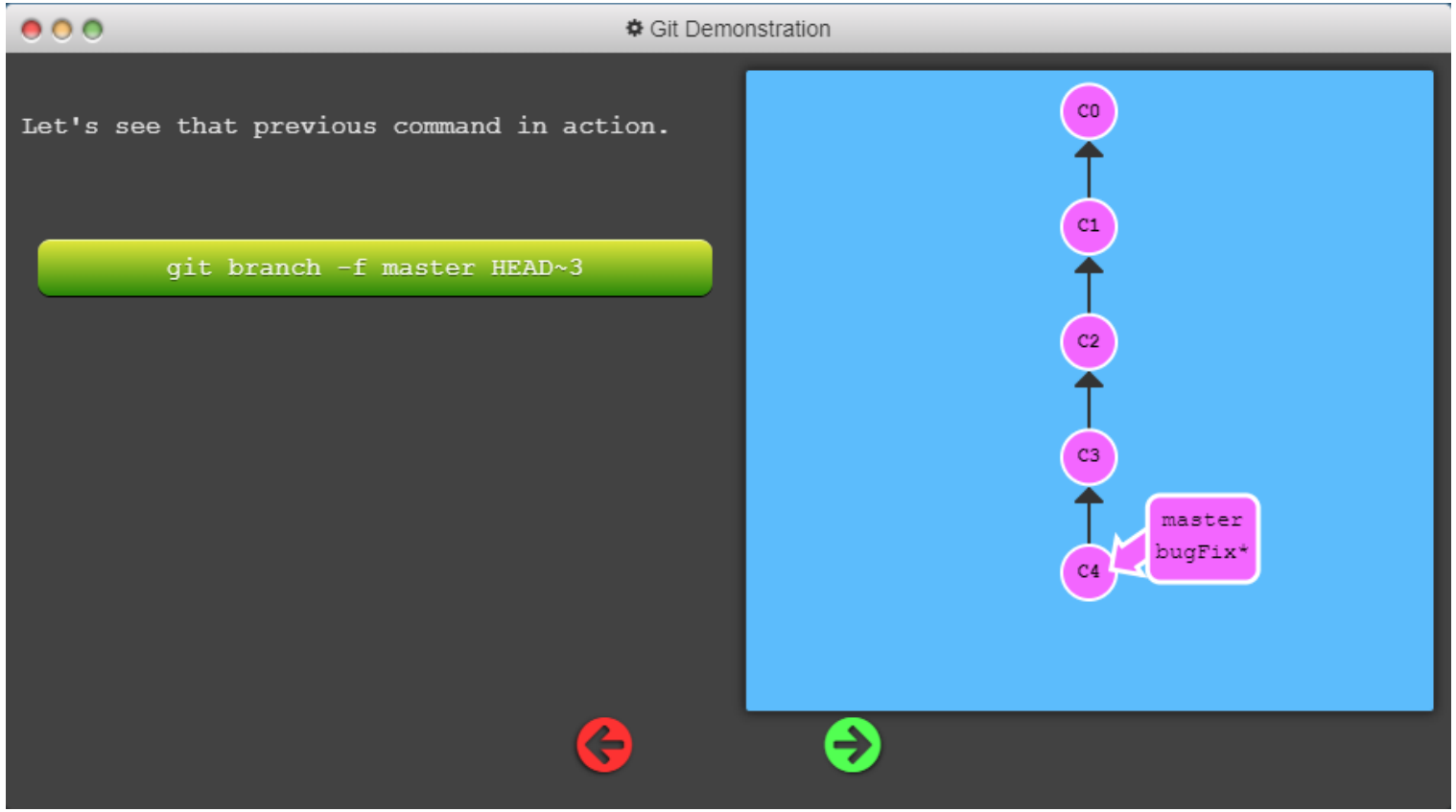


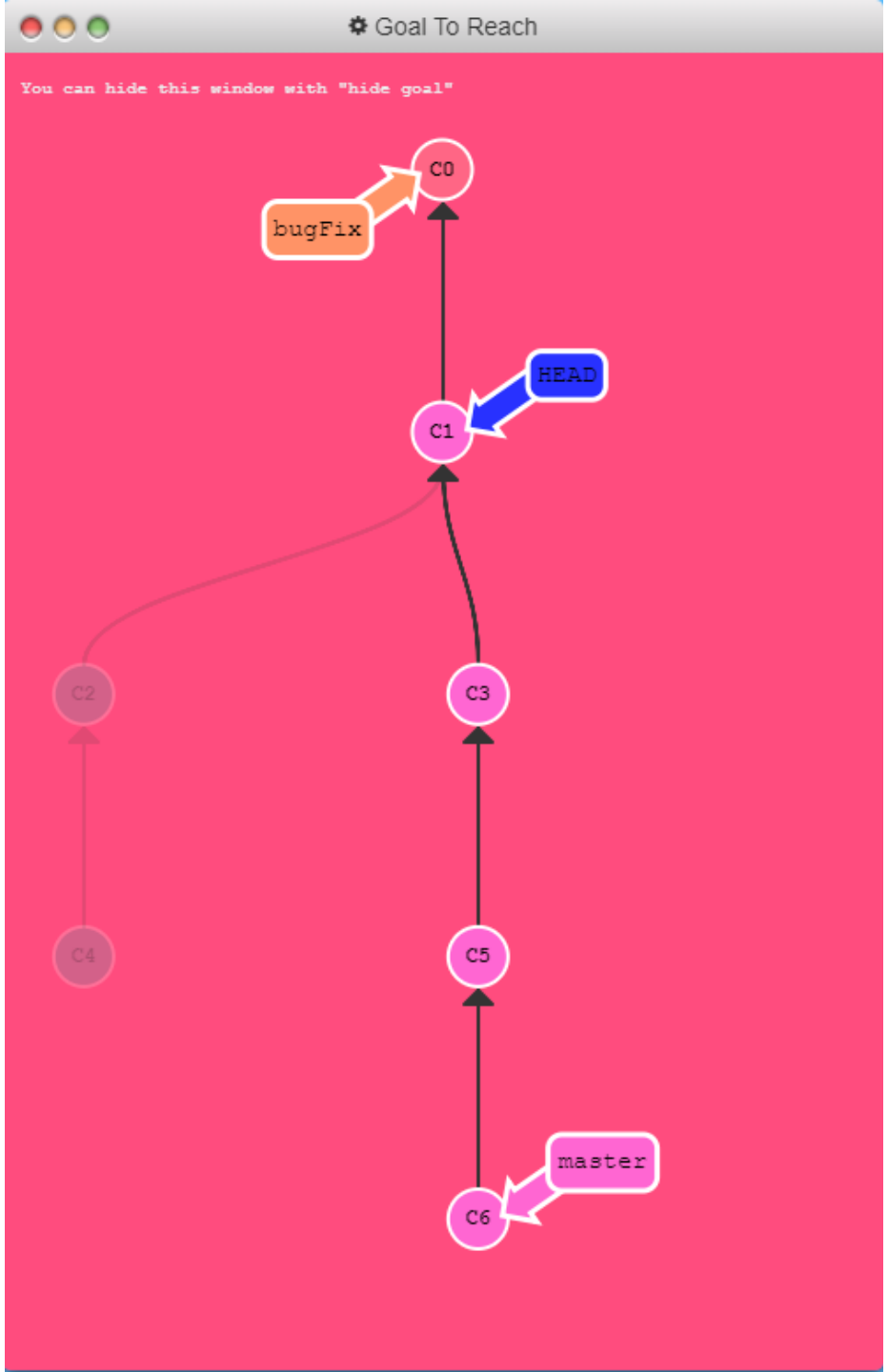
Homework #1 – Git Practice

- Branch forcing
  - Force branch changes

Homework #1 – Git Practice

$ git branch –f master HEAD~3



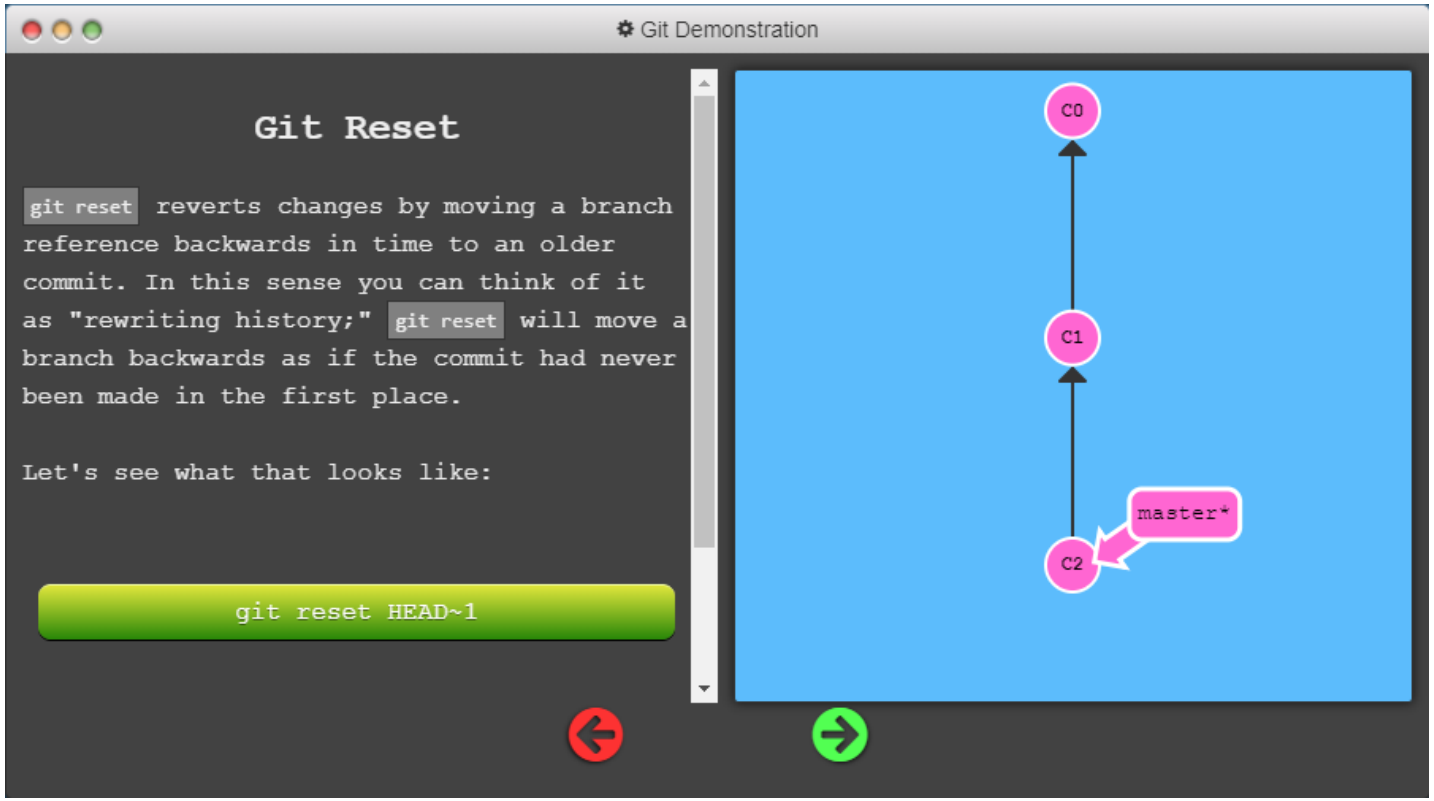

Goal To Reach
You can hide this window with "hide goal"
C0
bugFix
HEAD
C1
C2
C3
C4
C5
master
C6

Homework #1 – Git Practice

- Reversing changes
  - Cancel modification
  - Two ways to do so:
    - git reset
    - git revert

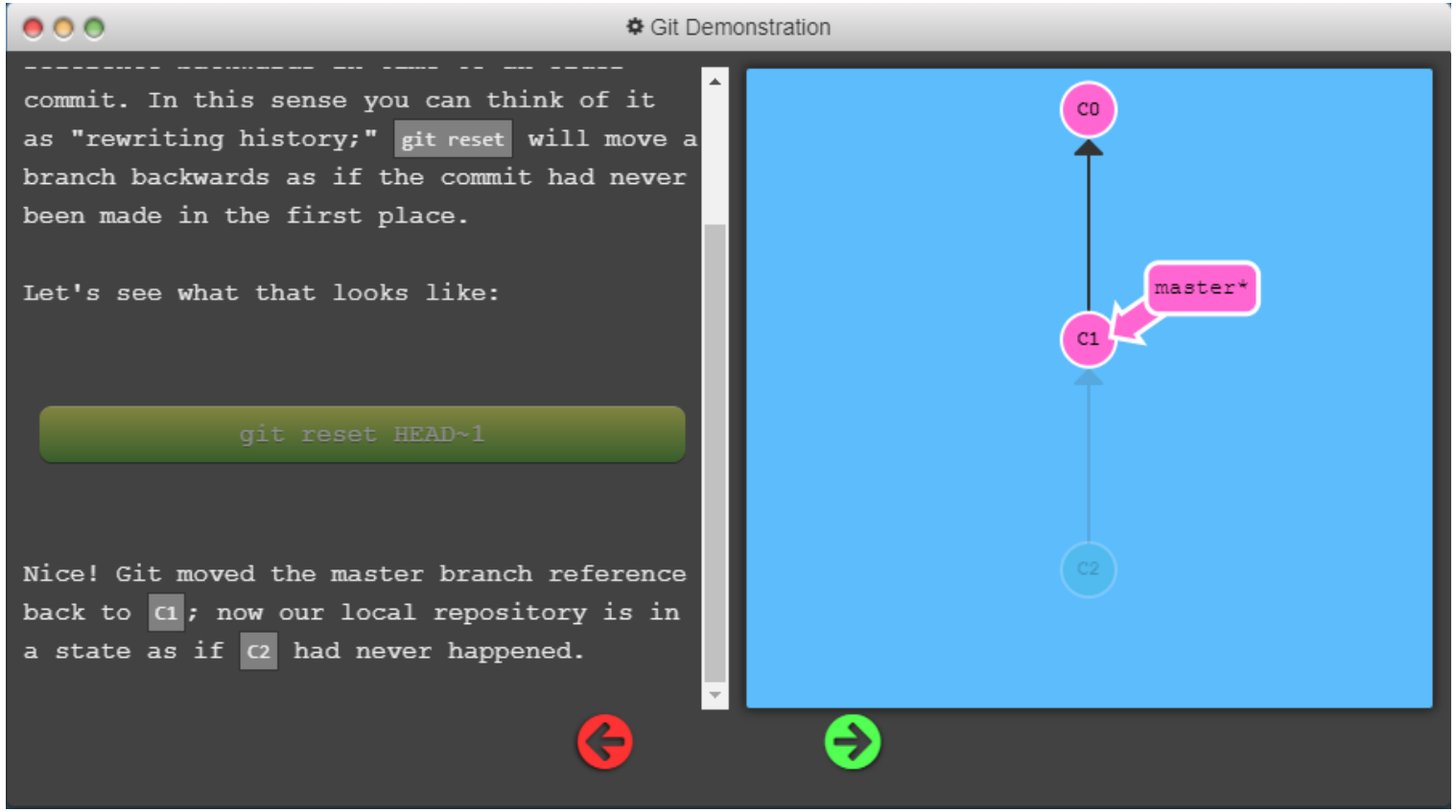
Homework #1 – Git Practice

$ git reset HEAD~1



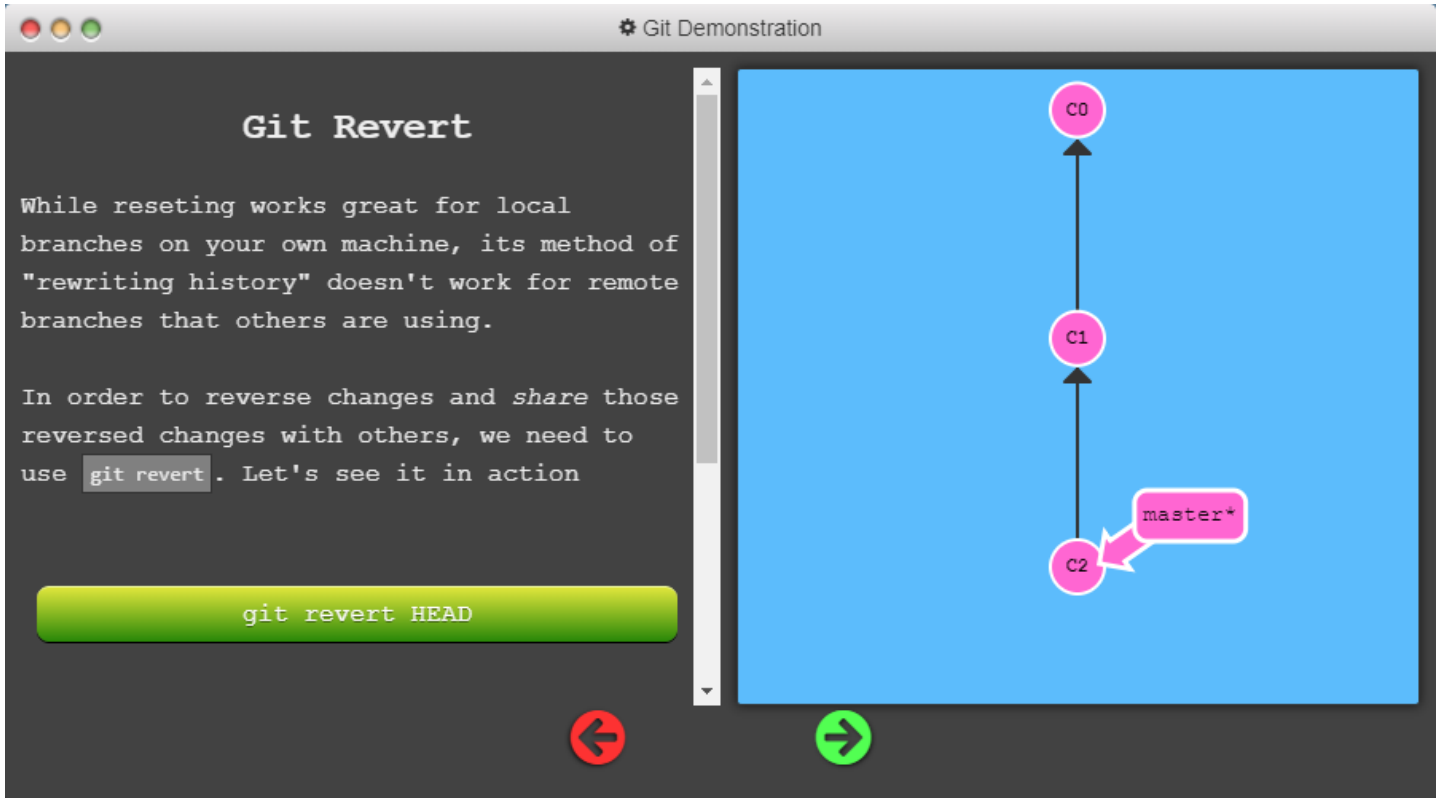
Homework #1 – Git Practice

$ git reset HEAD~1



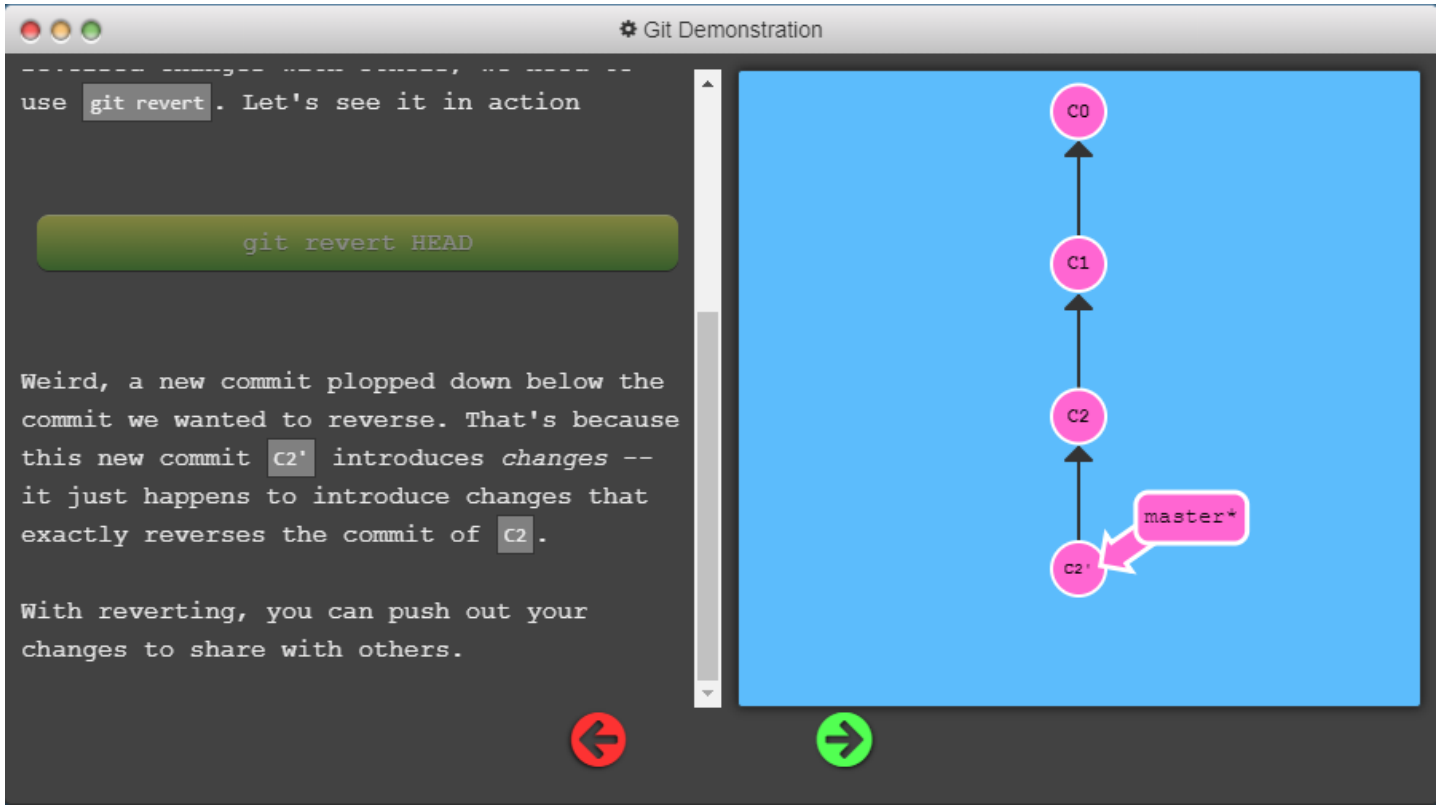
Homework #1 – Git Practice

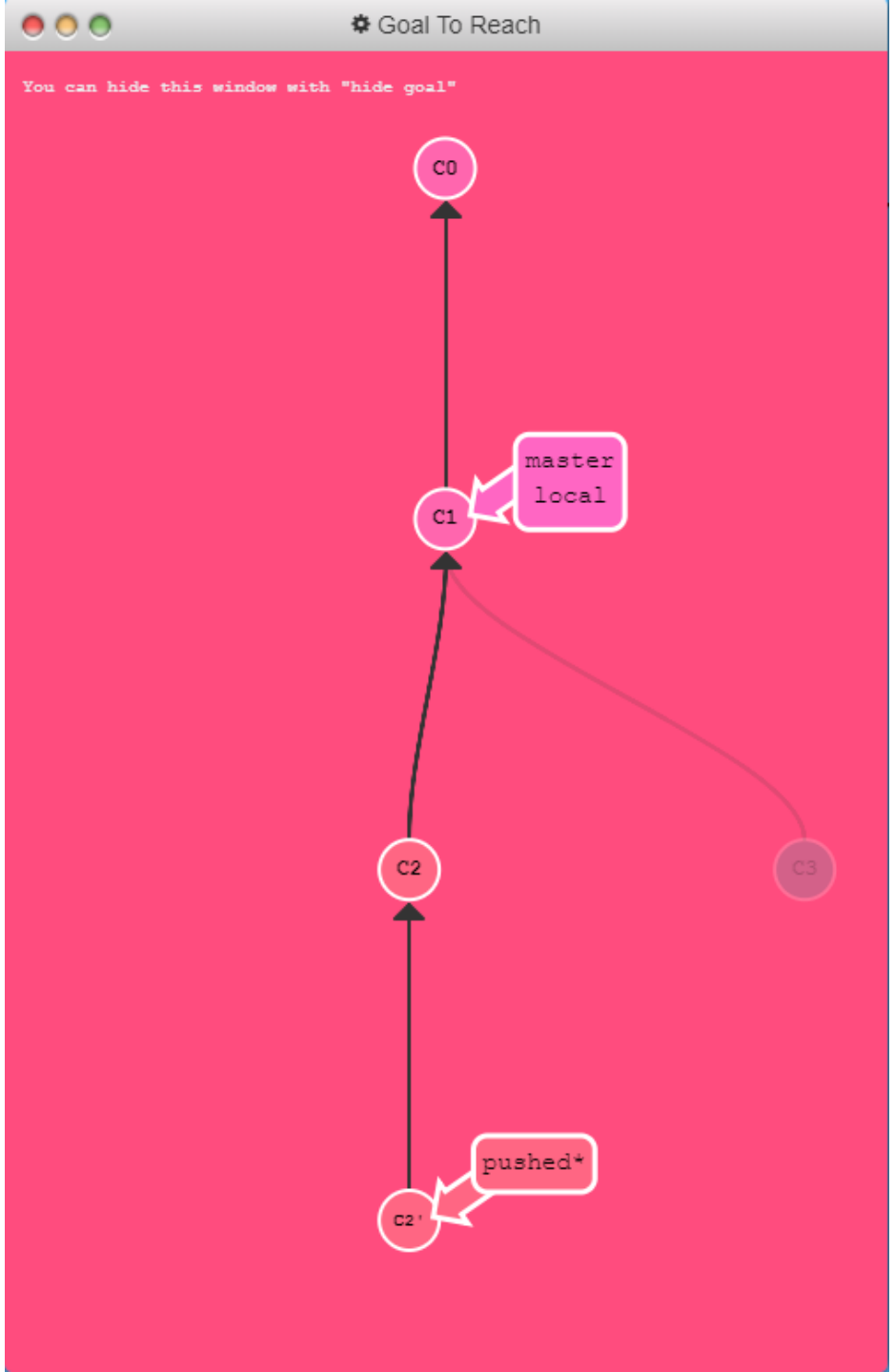
$ git revert HEAD



Homework #1 – Git Practice

$ git revert HEAD



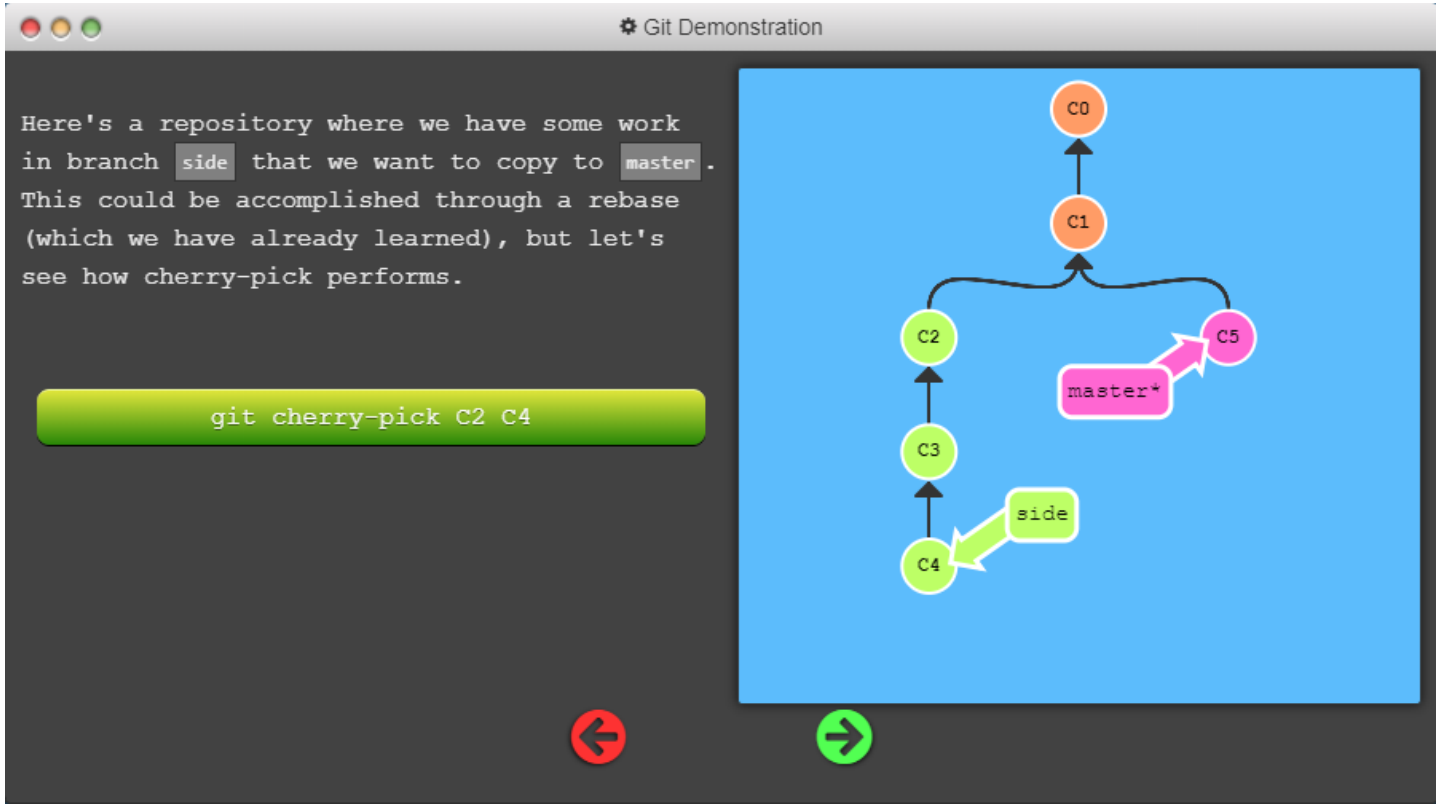

Goal To Reach
You can hide this window with "hide goal"
C0
master
local
C1
C2
C3
pushed*
C2'

Homework #1 – Git Practice

- Git Cherry-pick
  - Copy all the commits below the current commit (where the HEAD is located)

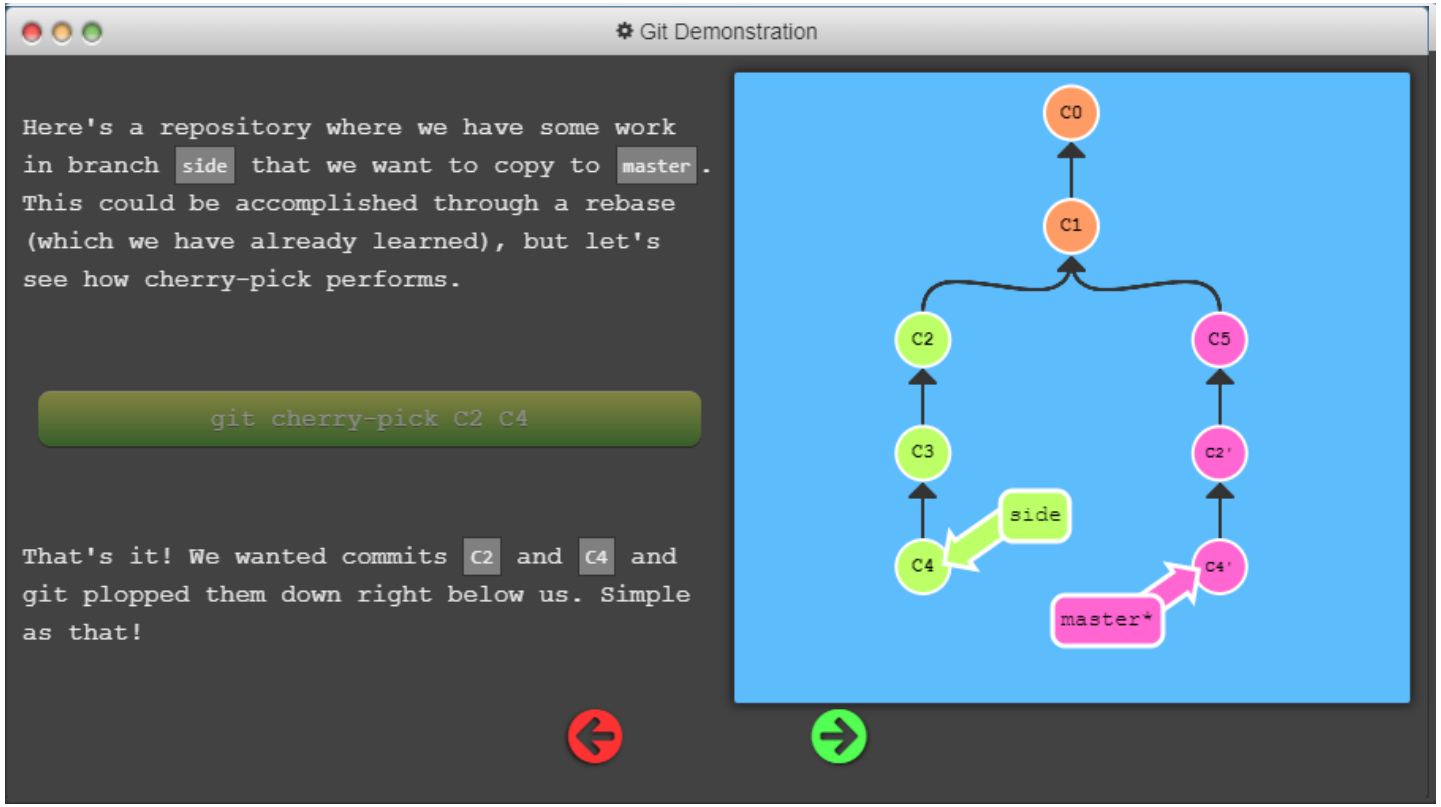
Homework #1 – Git Practice

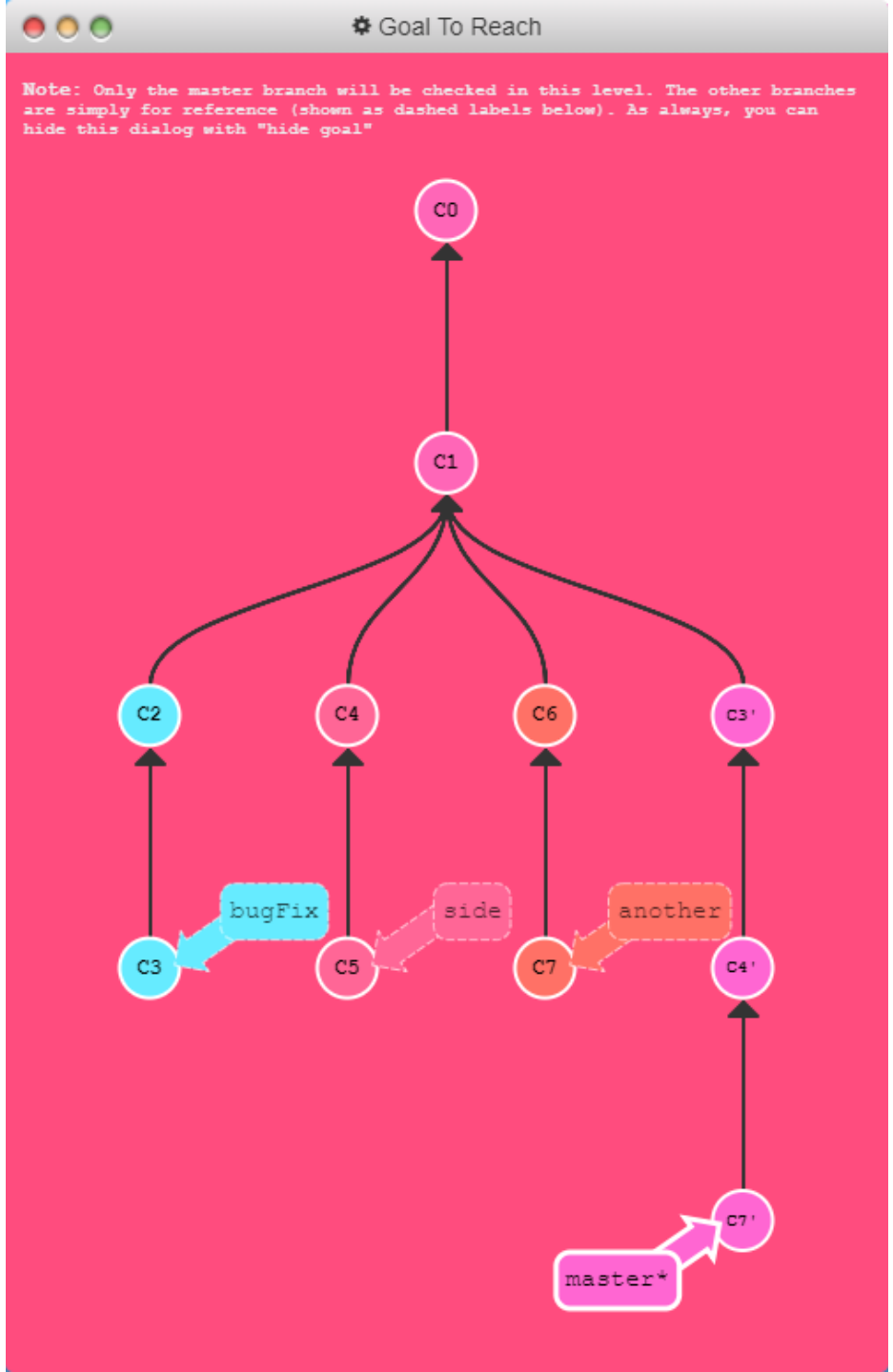
$ git cherry-pick C2 C4



Homework #1 – Git Practice

$ git cherry-pick C2 C4



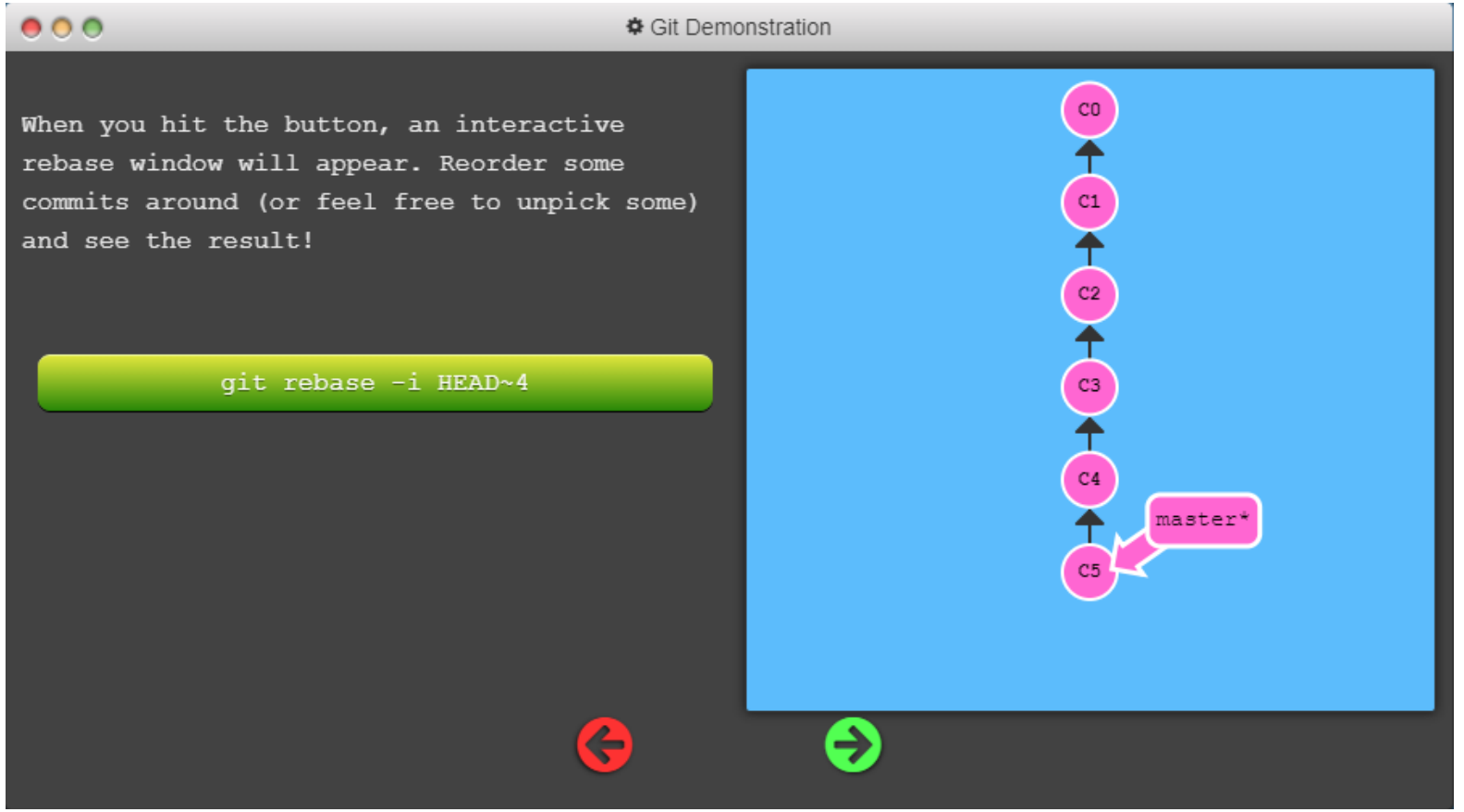

Goal To Reach
Note: Only the master branch will be checked in this level. The other branches are simply for reference (shown as dashed labels below). As always, you can hide this dialog with "hide goal"
C0
C1
C2
C4
C6
C3'
bugFix
side
another
C3
C5
C7
C4'
C7'
master*

Homework #1 – Git Practice

- Git Interactive Rebase
  - If you know which commit to copy, it's useful to use cherry-pick
  - If not, you can review commits to decide what to copy
  - Functionalities:
    - Change the order of the commits
    - Remove certain commits

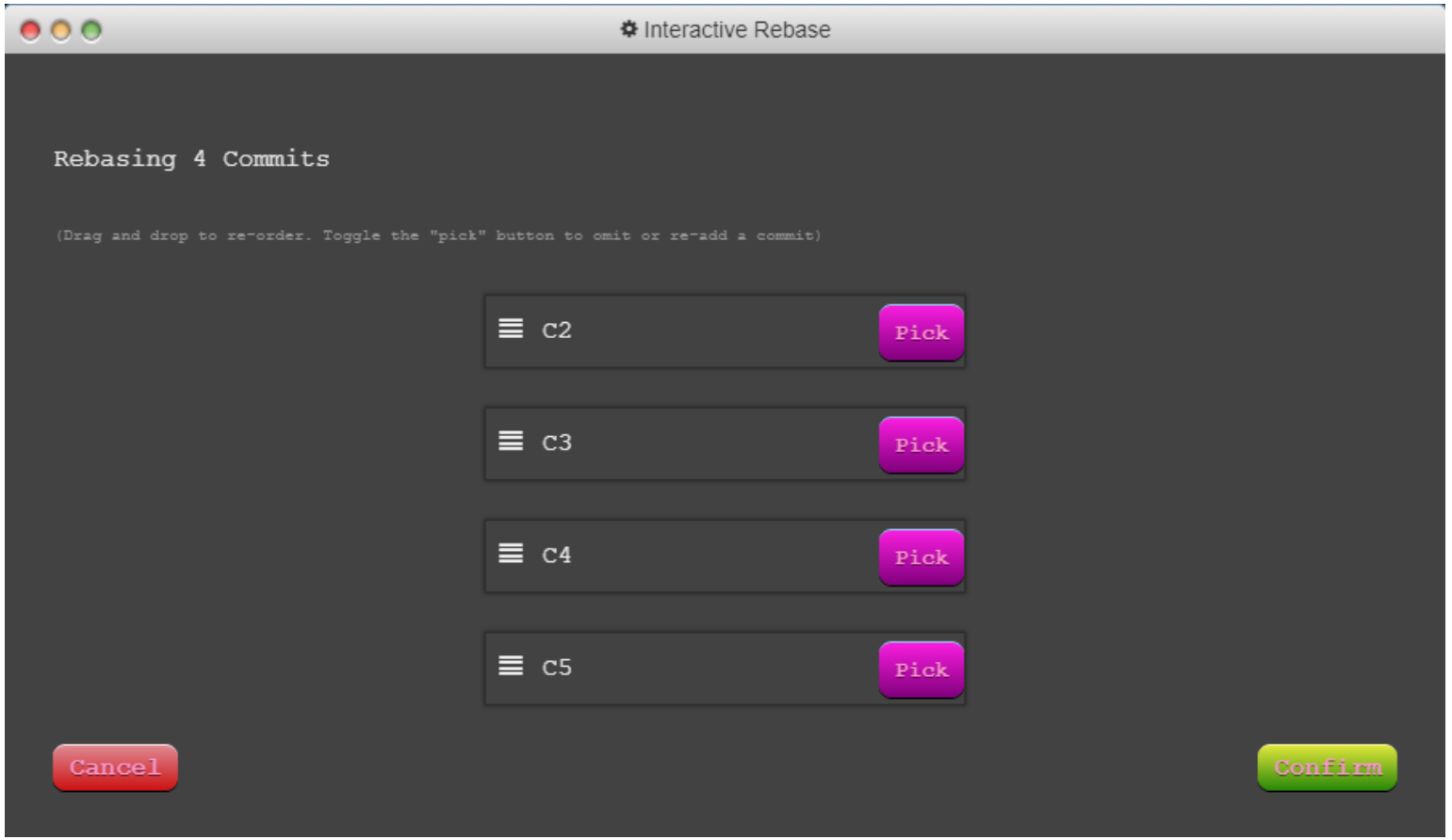
Homework #1 – Git Practice

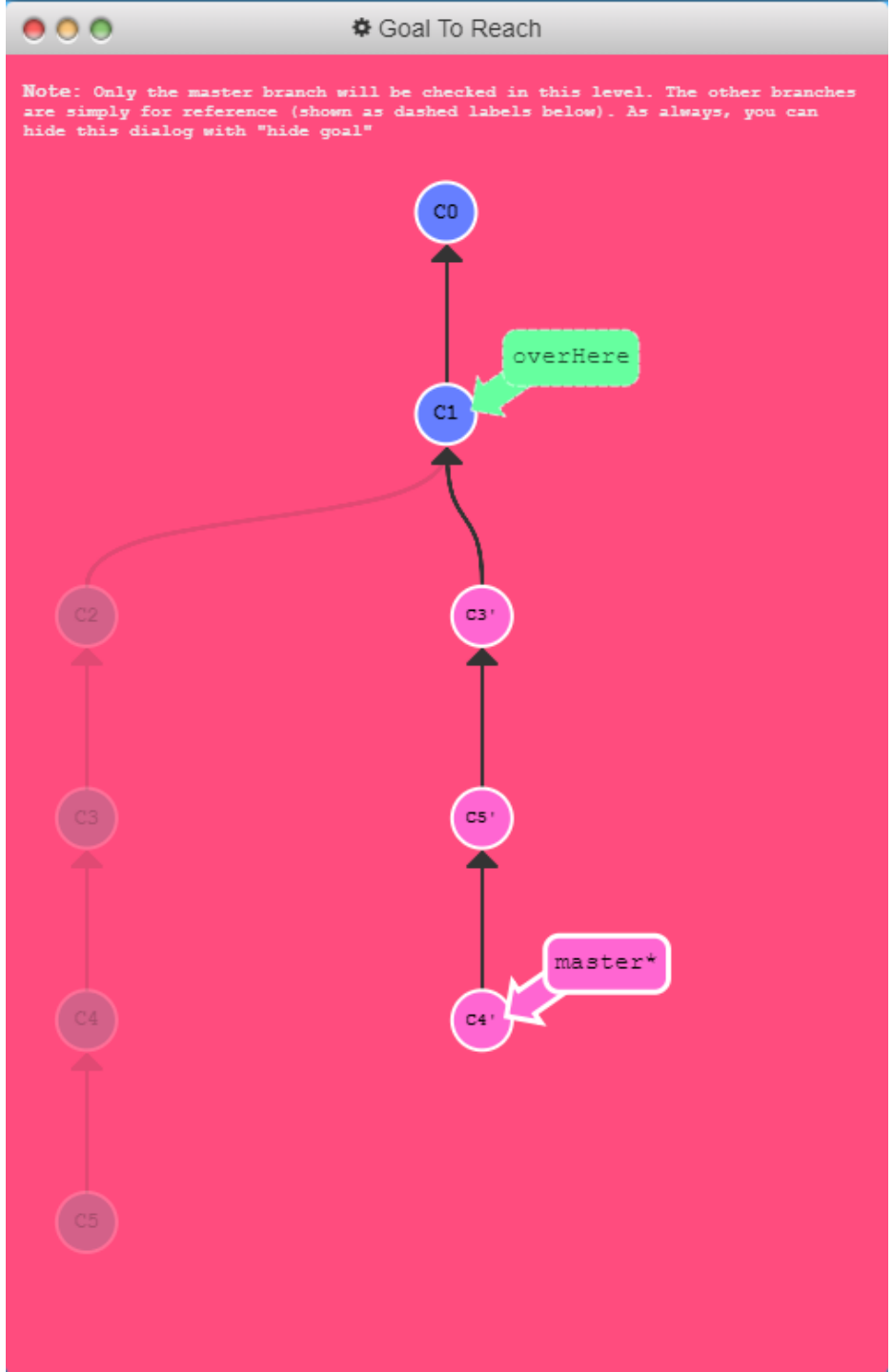
$ git rebase -i HEAD~4



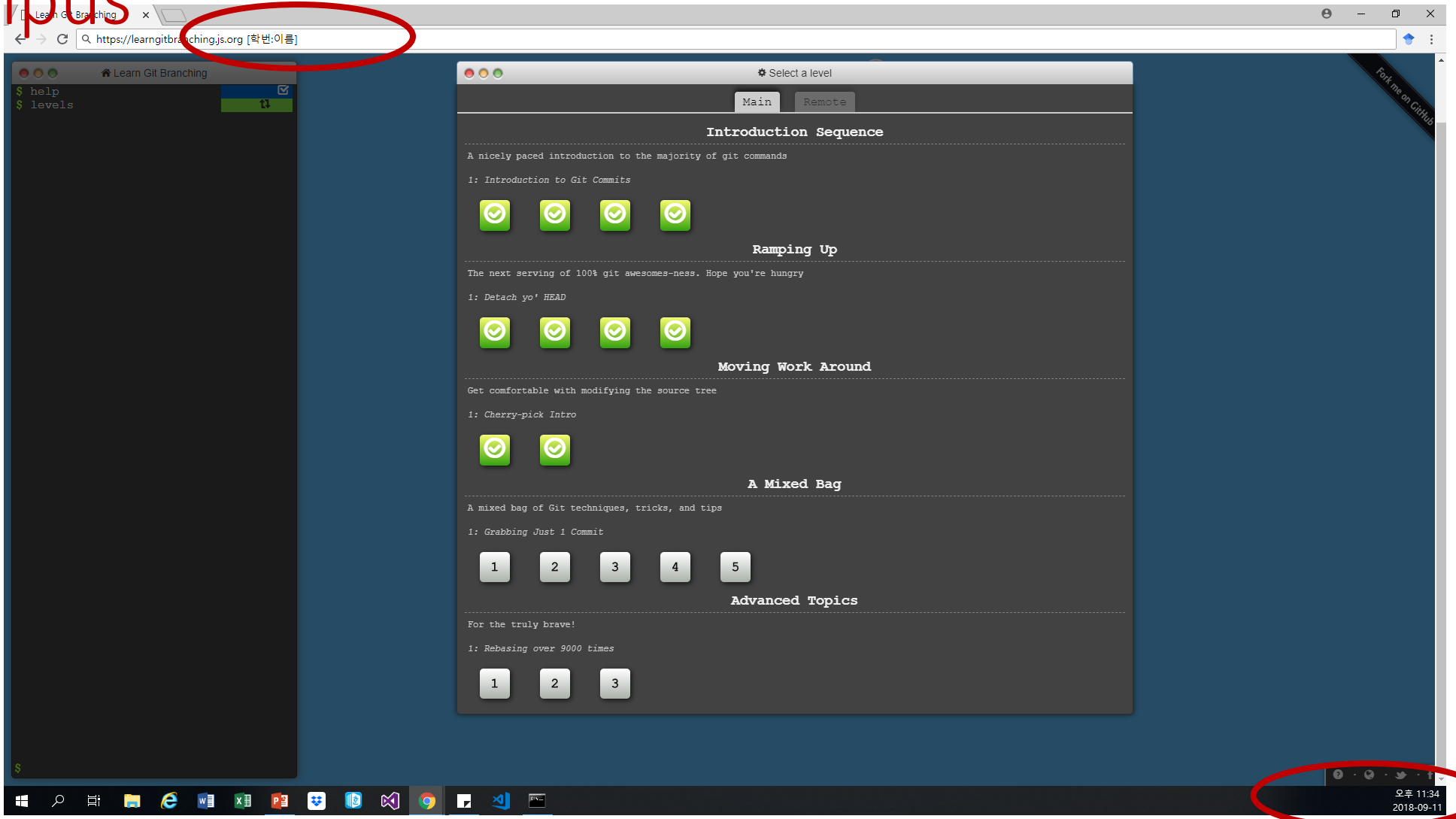
Homework #1 – Git Practice

$ git rebase -i HEAD~4




Goal To Reach
Note: Only the master branch will be checked in this level. The other branches are simply for reference (shown as dashed labels below). As always, you can hide this dialog with "hide goal"
C0
overHere
C1
C2
C3'
C3
C5'
master*
C4
C4'
C5

Submit a screenshot including your name and Student ID along with the date to the Cyber Campus

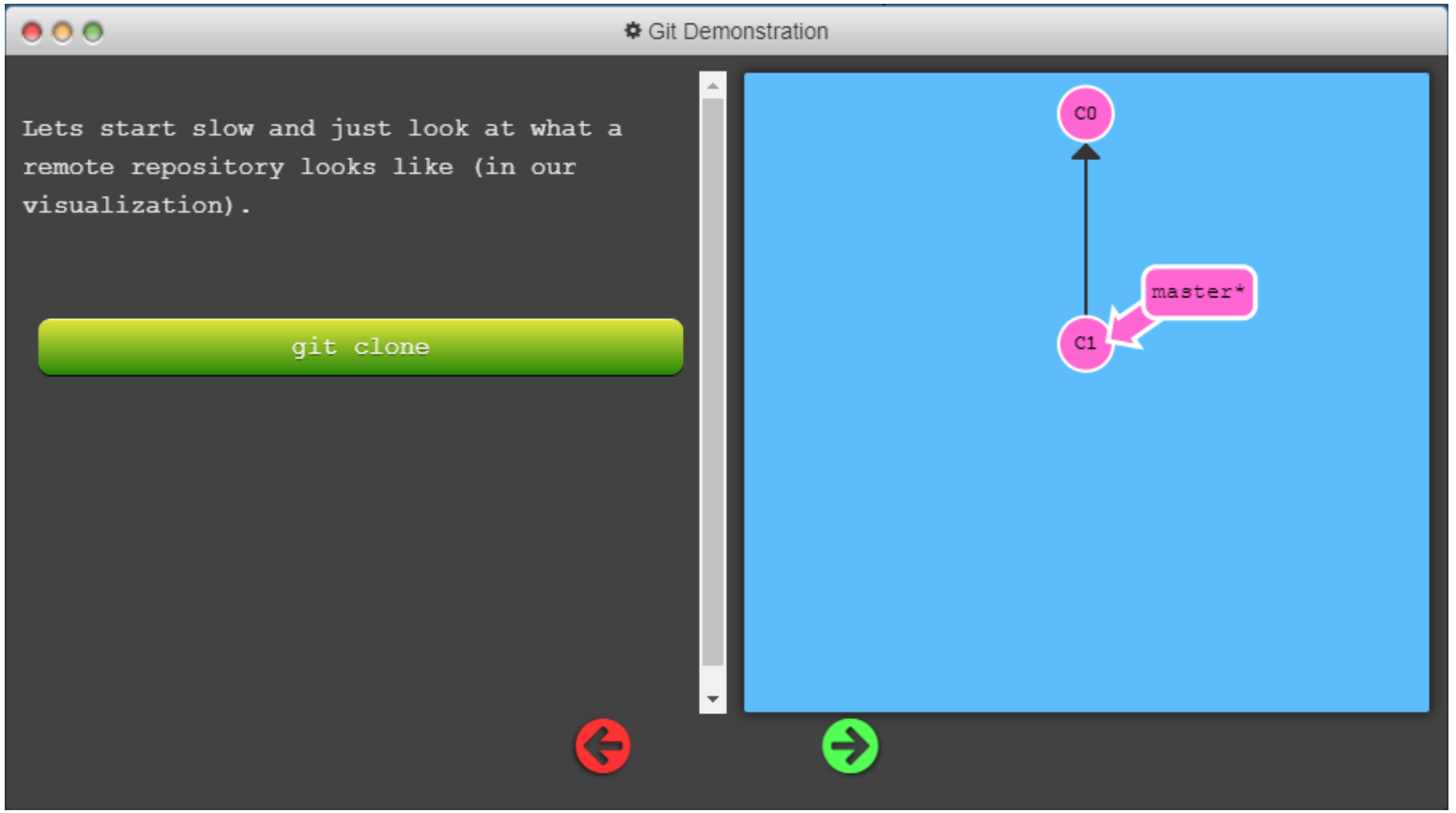


Homework #1.1 – Git Practice (remote) <- extra credit

- Clone Intro
  - Remote Repository:
    - Stored at another computer or on the web
      - For a backup
      - Or for collaboration (Social Coding)
- *git clone*
  - Creating a local copy from a remote repository

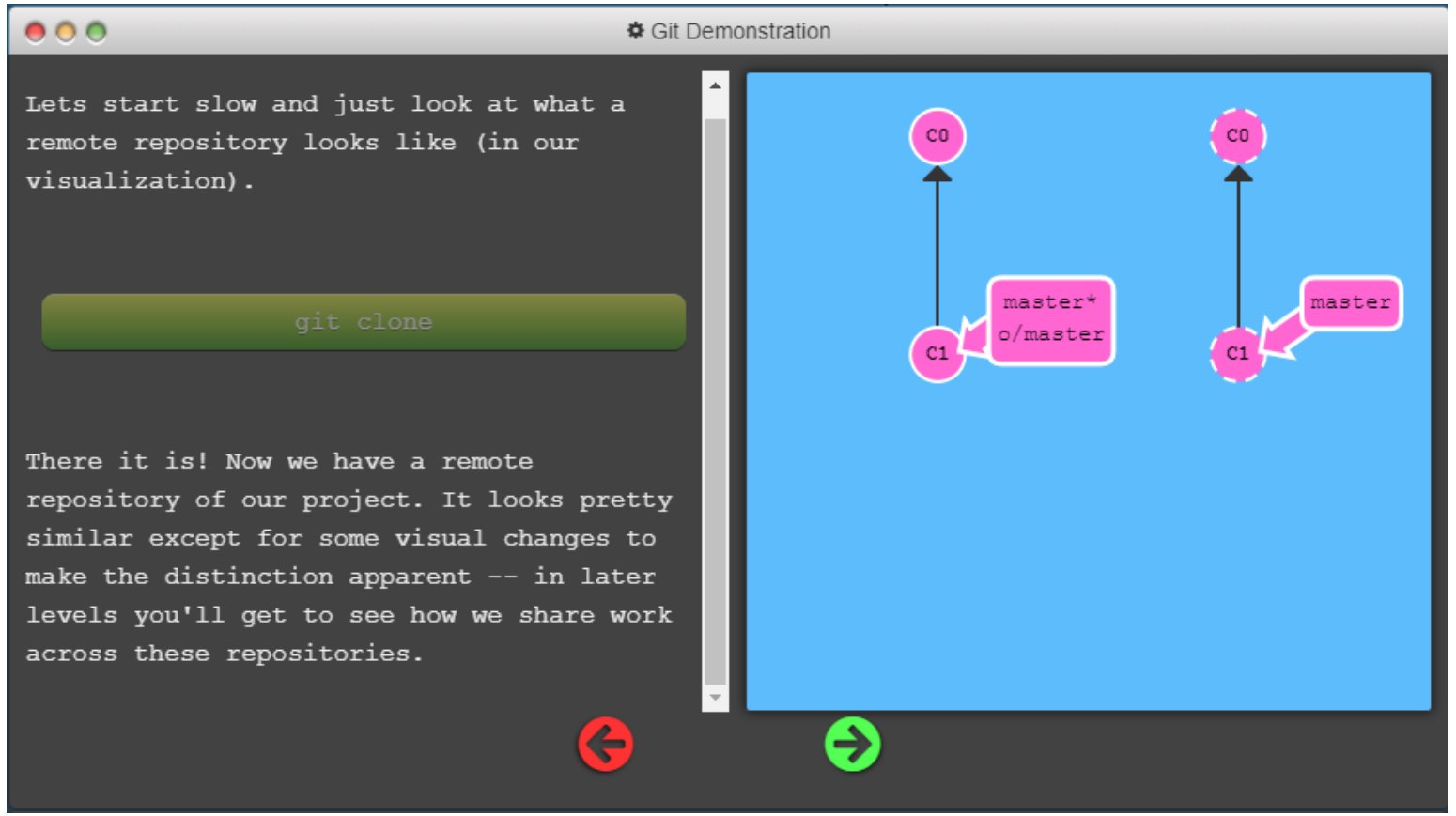
Homework #1.1 – Git Practice

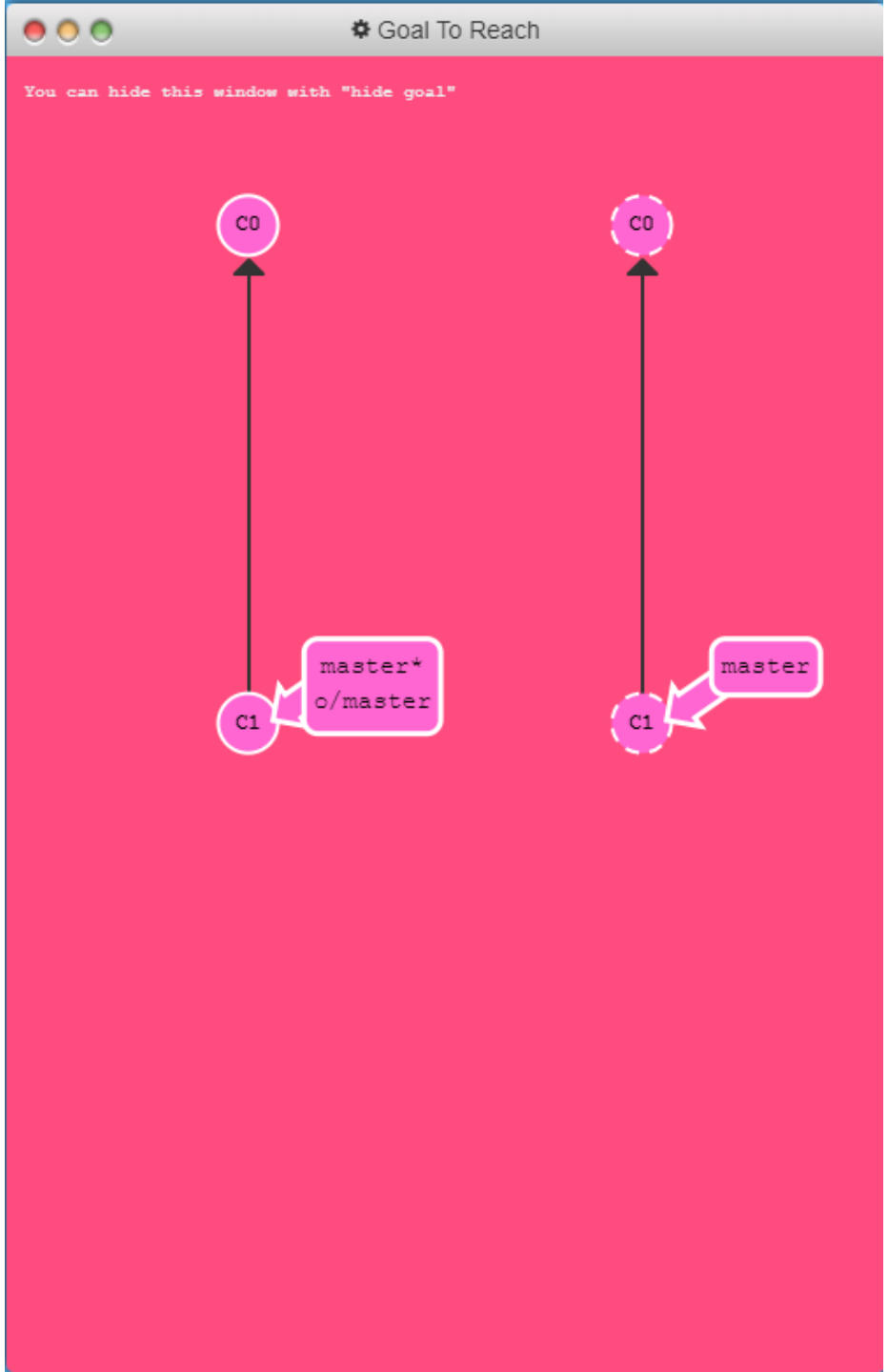
$ git clone



Homework #1.1 – Git Practice

$ git clone



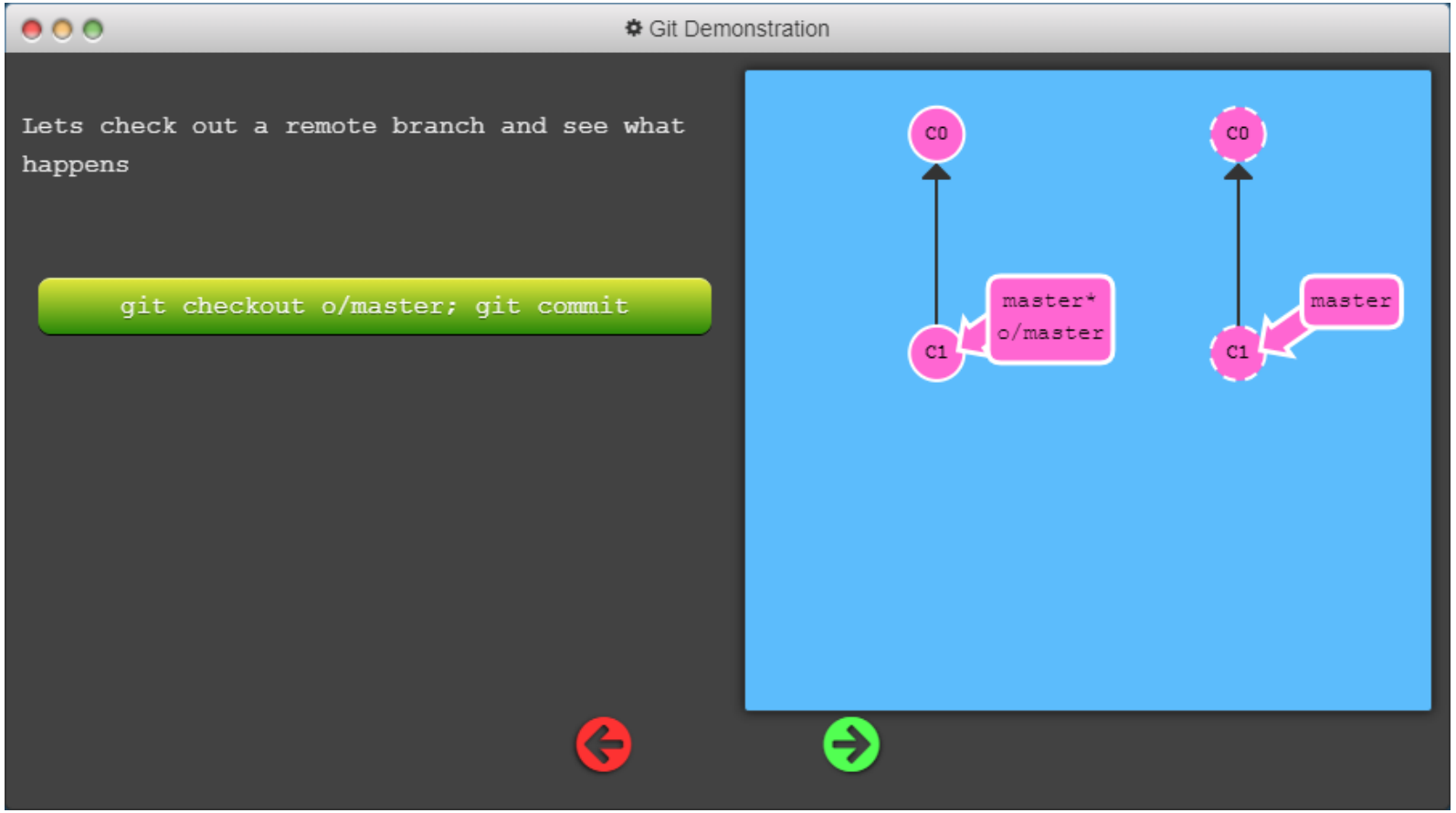

Goal To Reach
You can hide this window with "hide goal"
C0
C1
master*
o/master
C0
C1
master

Homework #1.1 – Git Practice

- Remote Branch
  - Cannot make direct changes
  - But you can create a local branch and make changes to that branch and then update the remote branch with yours
  - You can see what has changed
  - <remote name>/<branch name>
    - Ex) origin/master

Homework #1.1 – Git Practice

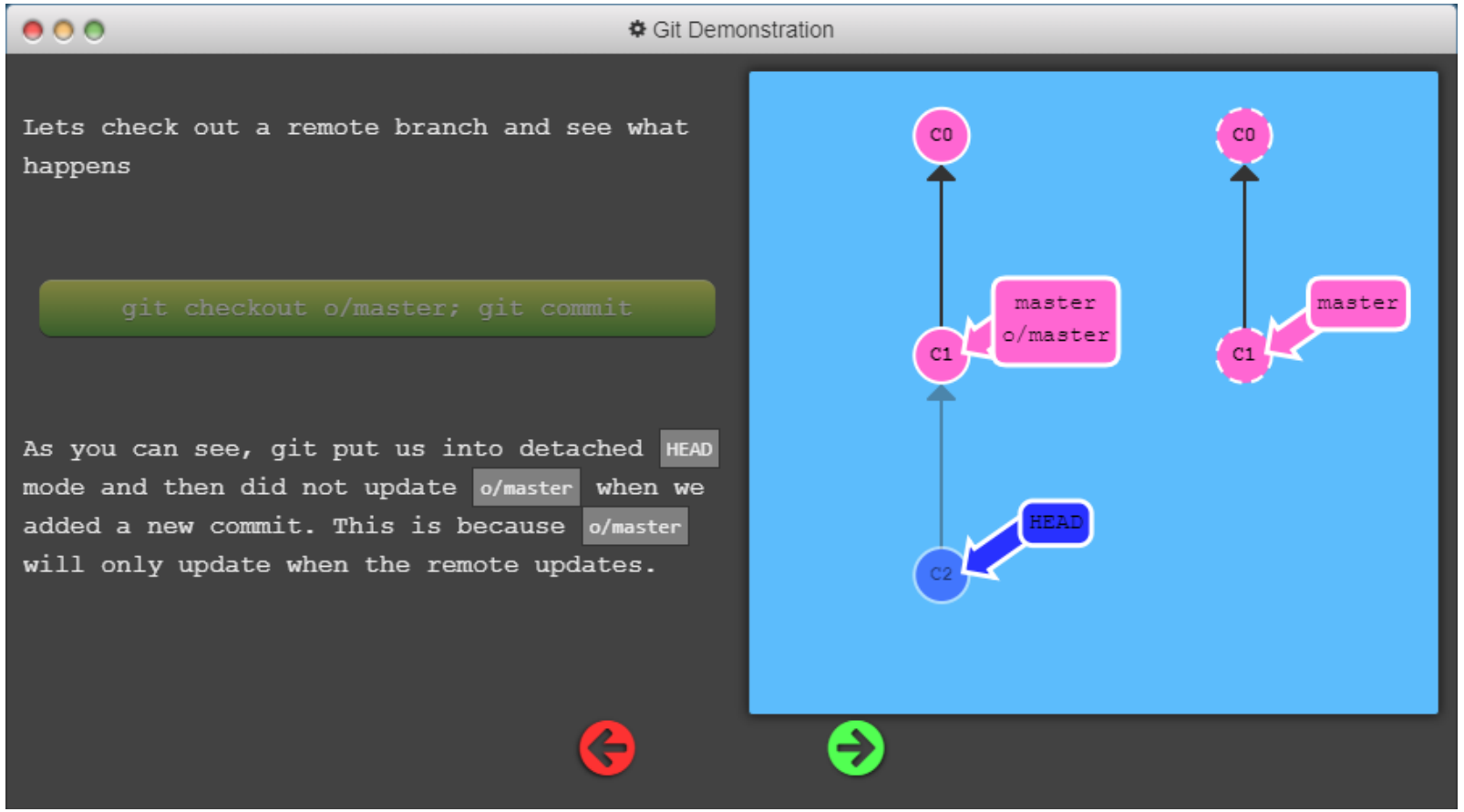
$ git checkout o/master; git commit

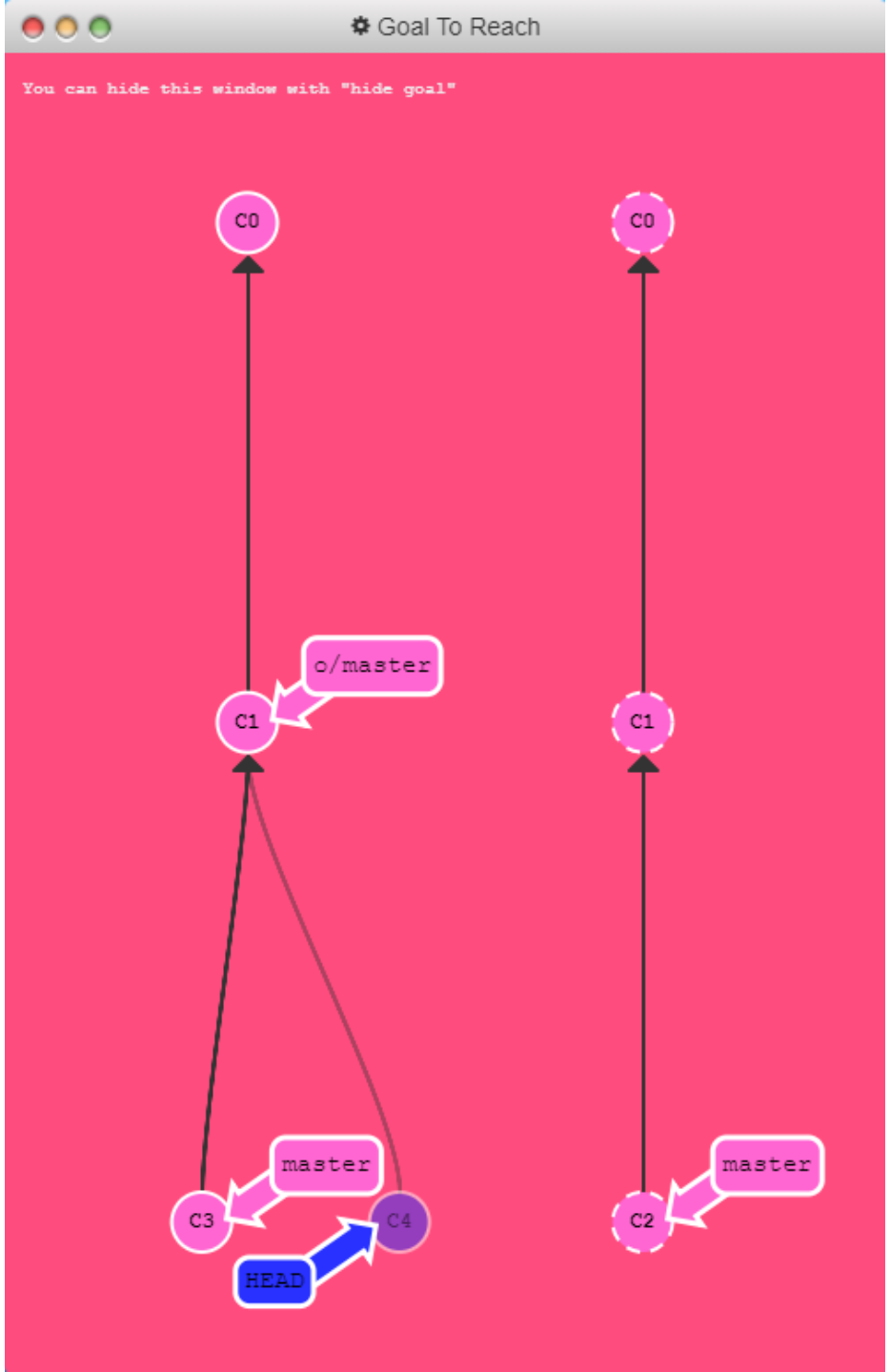


- Note that o/master is used instead of origin/master and that o/master is used as a local repository in this exercise

Homework #1.1 – Git Practice

$ git checkout o/master; git commit



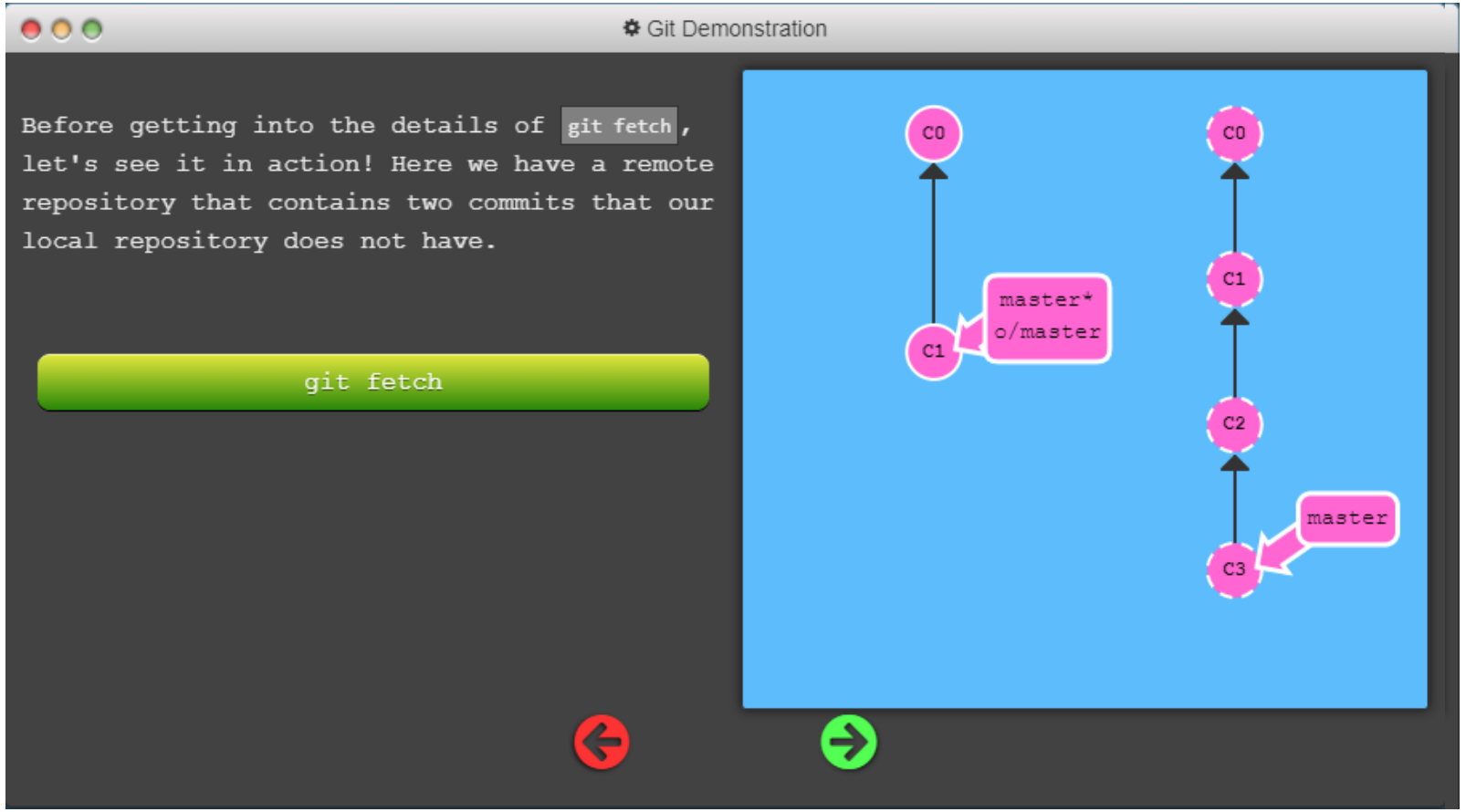

Goal To Reach
You can hide this window with "hide goal"
C0
C0
o/master
C1
C1
master
master
C3
C4
C2
HEAD

Homework #1.1 – Git Practice

- Git Fetchin'
  - It is very frequent to send and receive data to and from the remote repository
  - *git fetch*: it gets data from a remote repository and saves it to your local repository =
    - Step 1: download commits of the remote repository that are missing from the local repository
    - Step 2: update the pointer for the remote repository
  - Note that it is downloading the missing files only, it does not affect your local repository
    - Your pointer remains the same

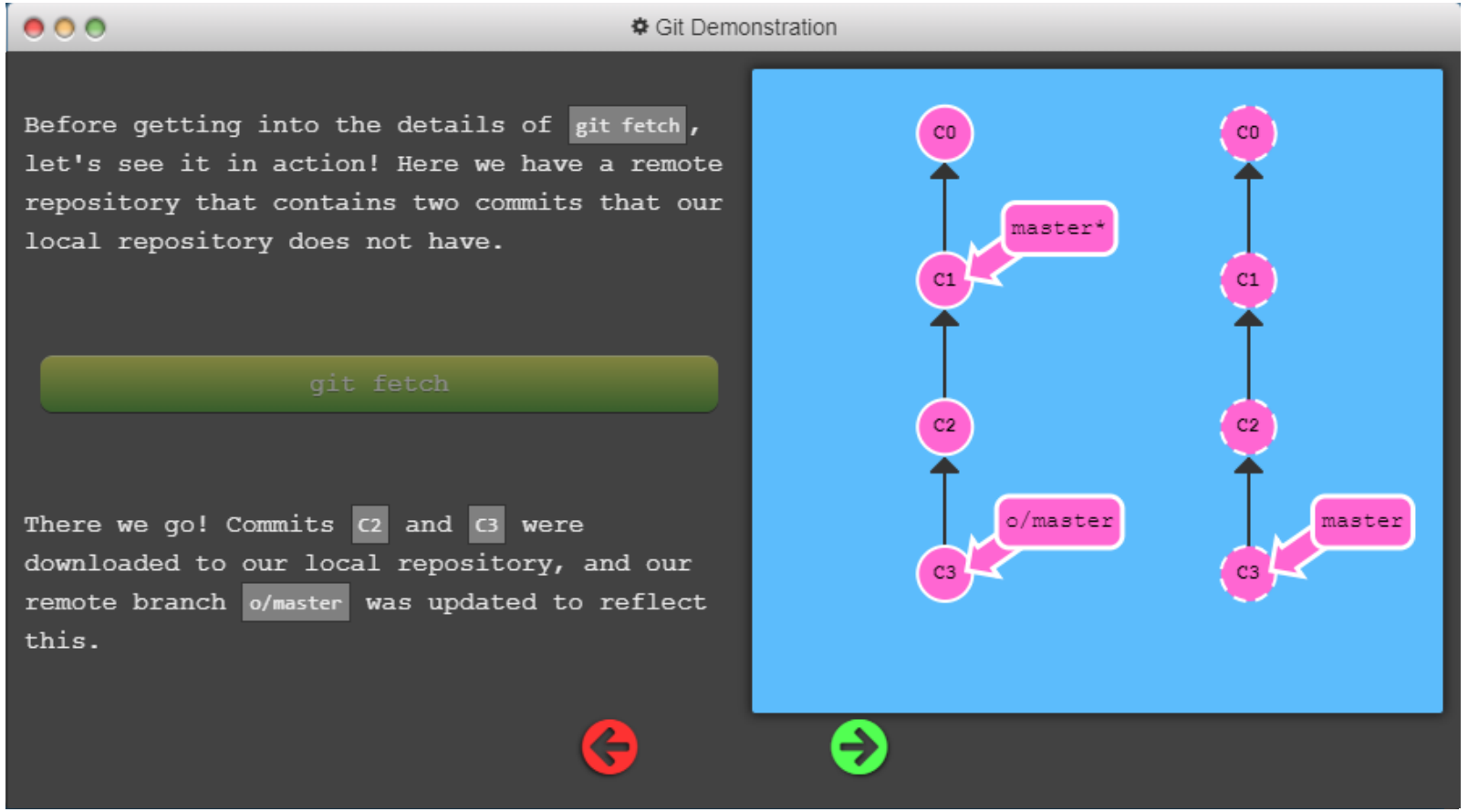
Homework #1.1 – Git Practice

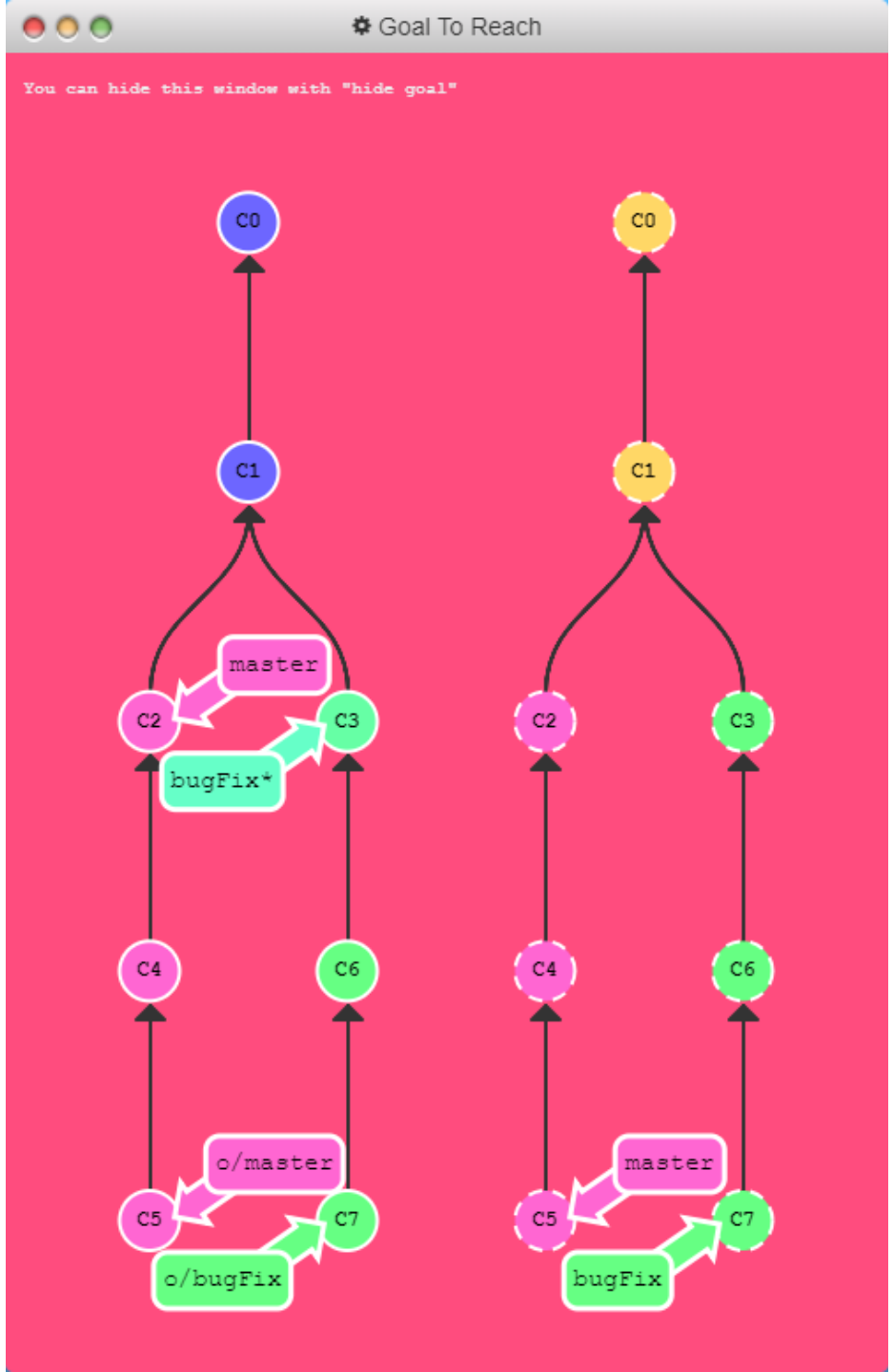
$ git fetch



Homework #1.1 – Git Practice

$ git fetch



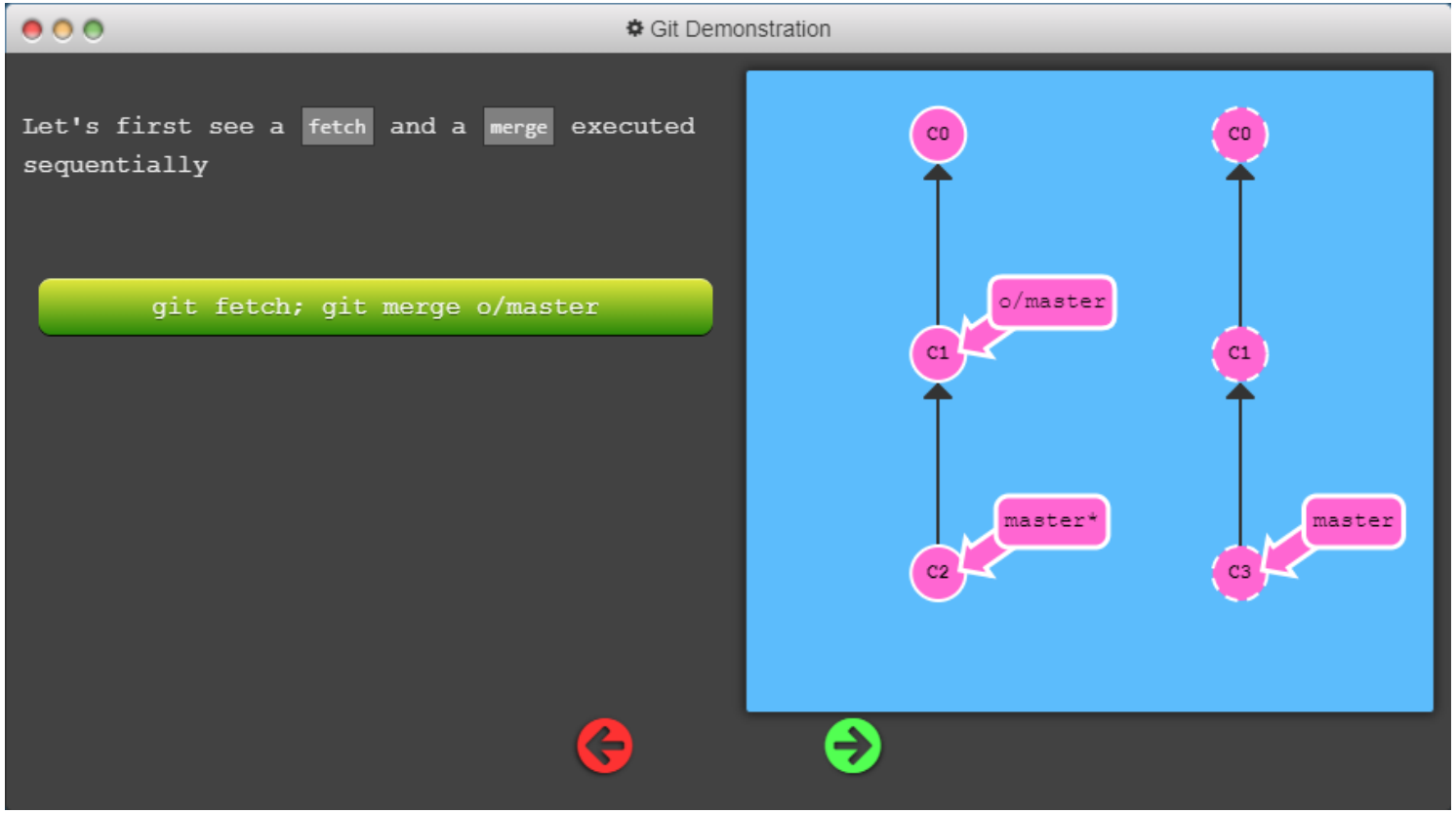

Goal To Reach
You can hide this window with "hide goal"
C0
C1
master
C2
C3
bugFix*
C4
C6
o/master
C5
C7
o/bugFix
C0
C1
C2
C3
C4
C6
master
C5
C7
bugFix

Homework #1.1 – Git Practice

- Git Pullin '
  - Updating your local repository once fetched
  - Ways to do so:
    - git cherry-pick o/master
    - git rebase o/master
    - git merge o/master
  - *git pull*: fetching & merging at the same time

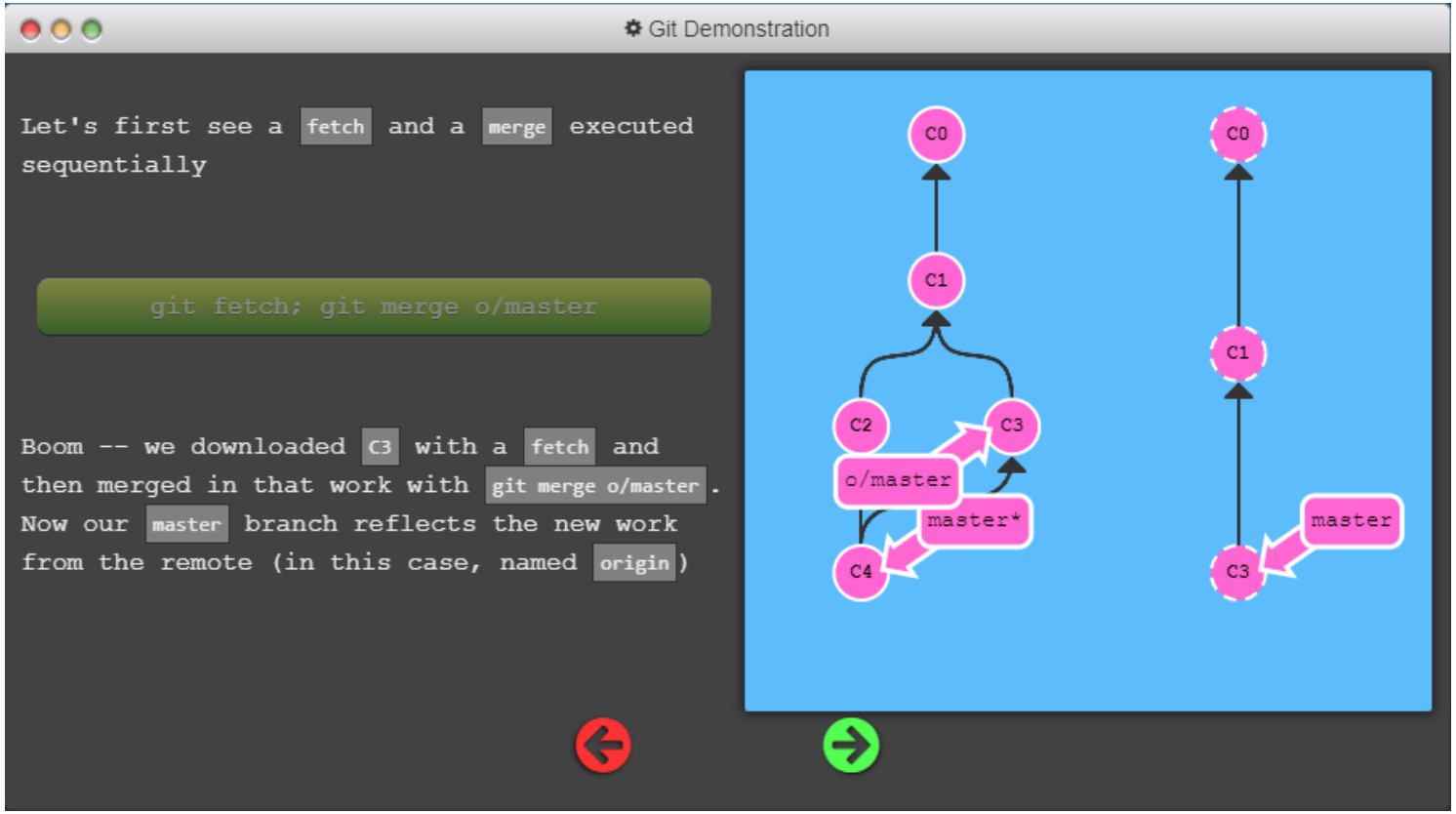
Homework #1.1 – Git Practice

$ git fetch; git merge o/master



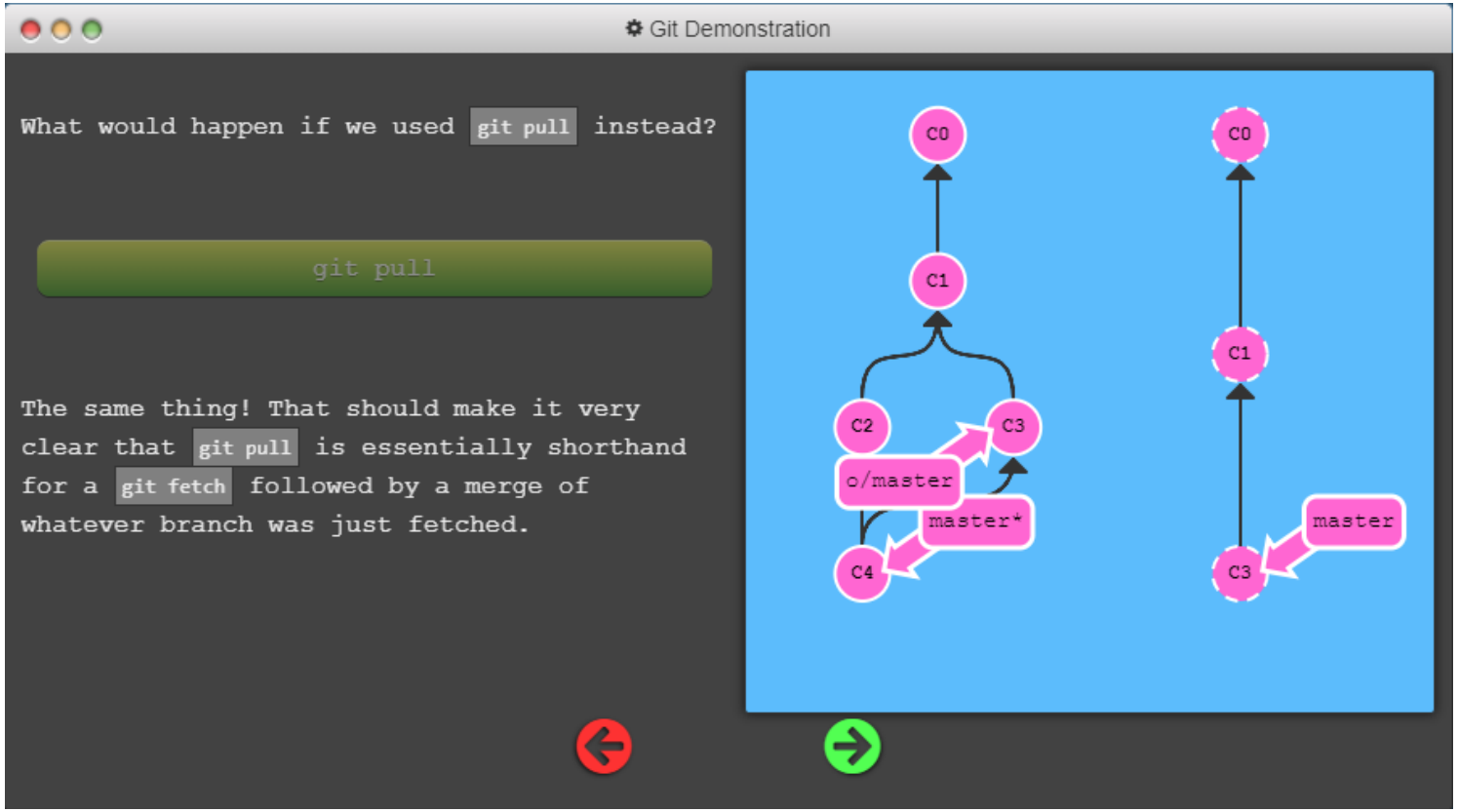
Homework #1.1 – Git Practice

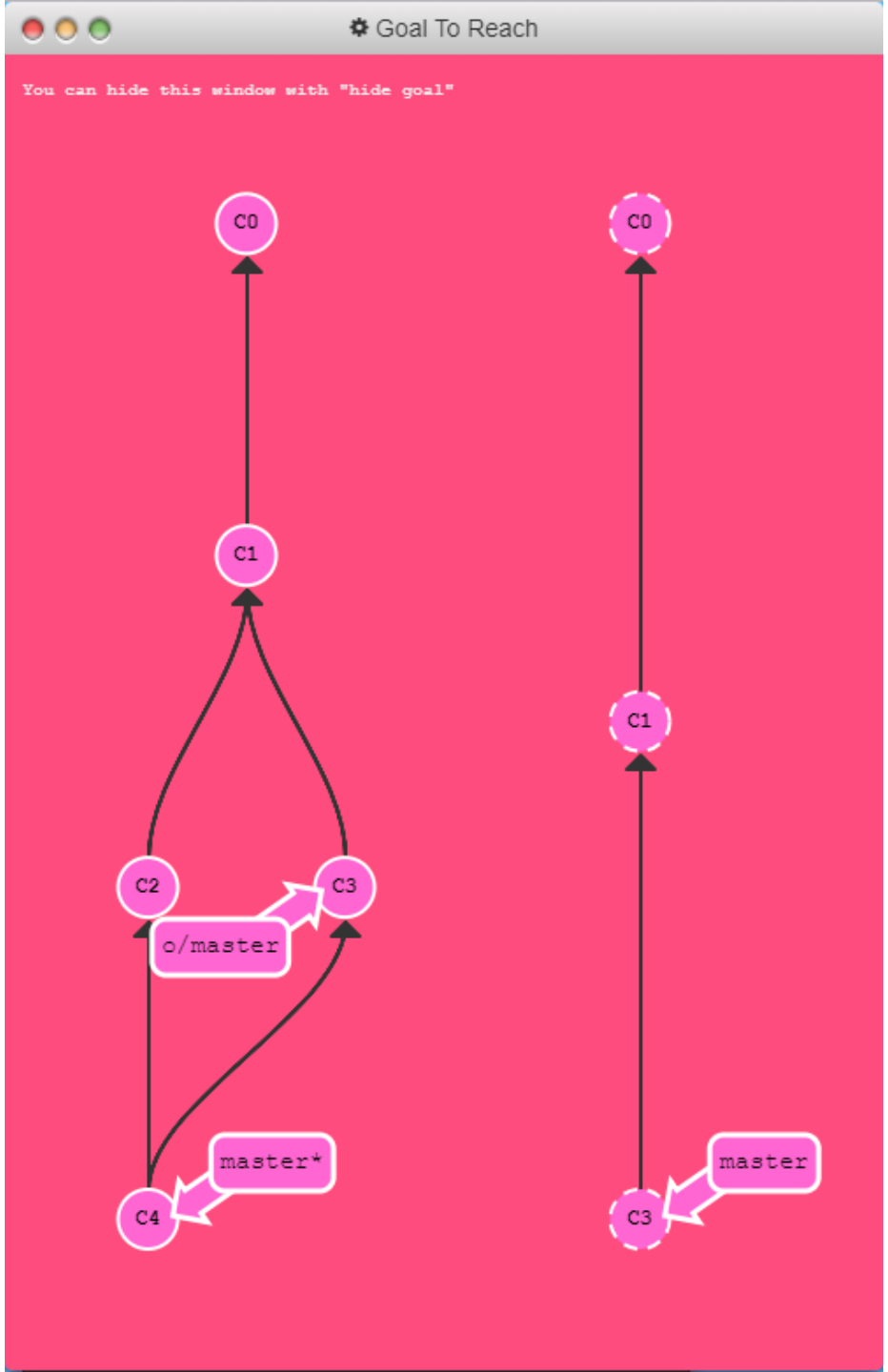
$ git fetch; git merge o/master



Homework #1.1 – Git Practice

$ git pull



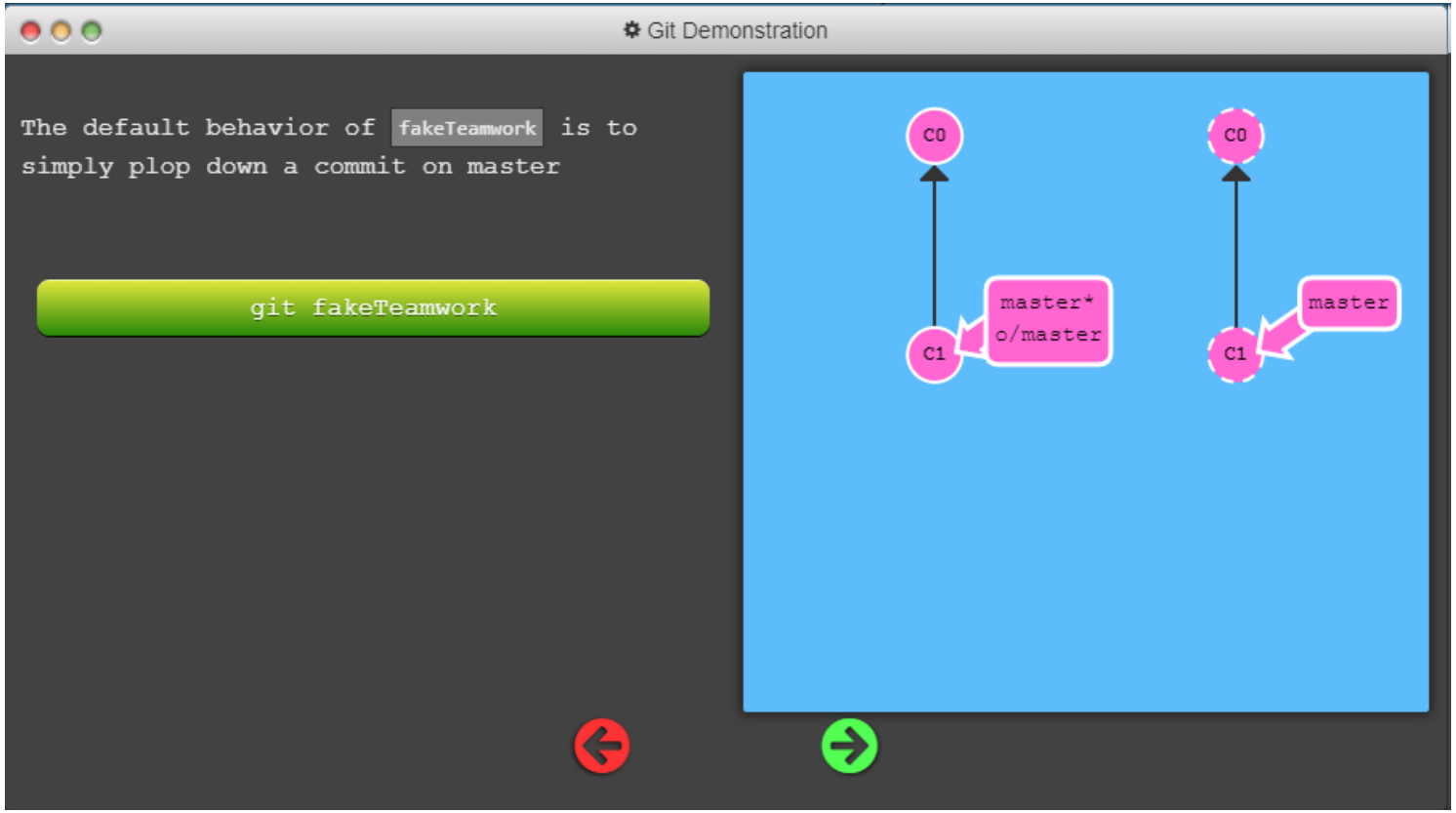

Goal To Reach
You can hide this window with "hide goal"
C0
C1
C2
C3
o/master
master*
C4
C0
C1
master
C3

Homework #1.1 – Git Practice

- Faking Teamwork
  - Experiencing collaboration with other developers

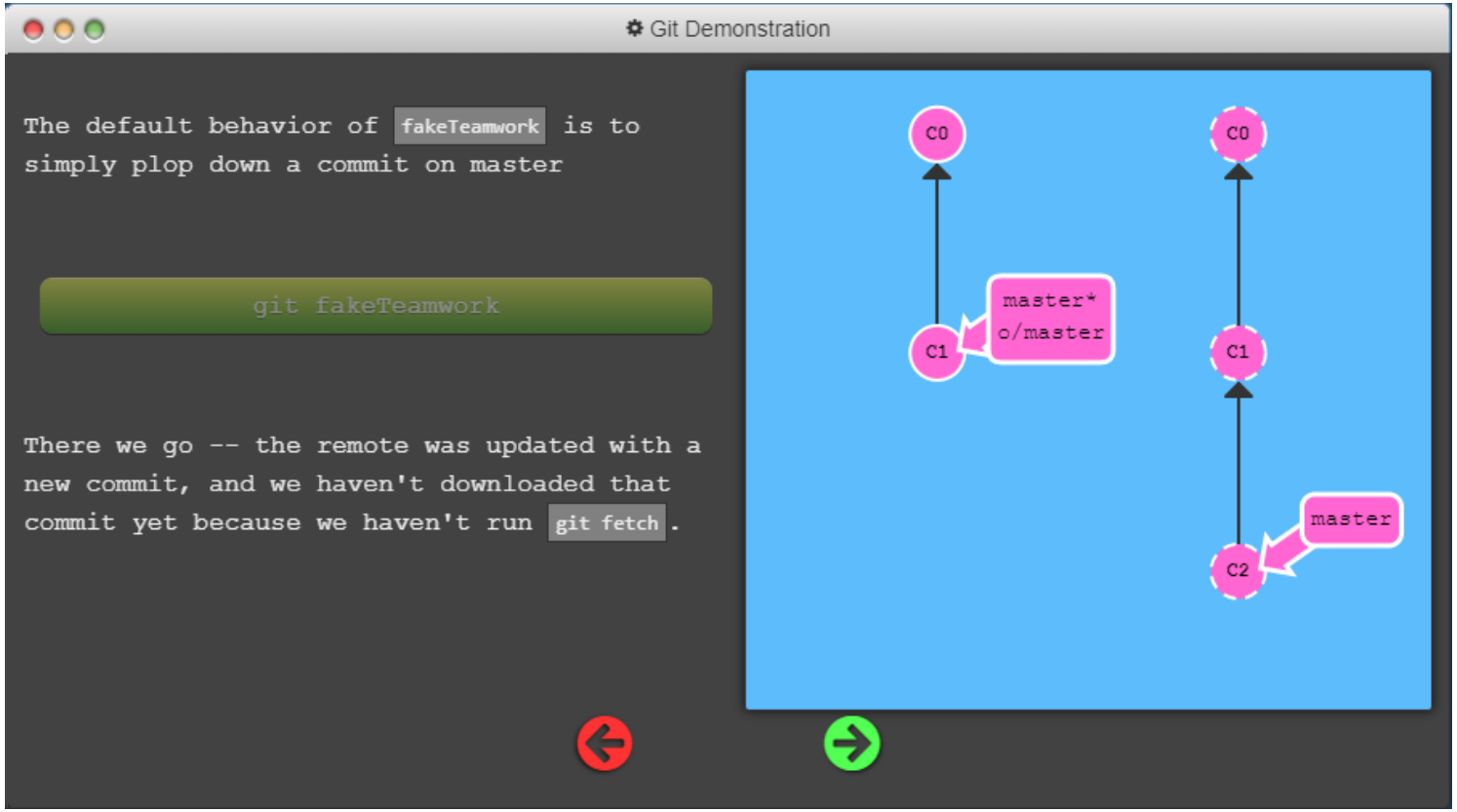
Homework #1.1 – Git Practice

$ git fakeTeamwork (<- Only for this exercise)



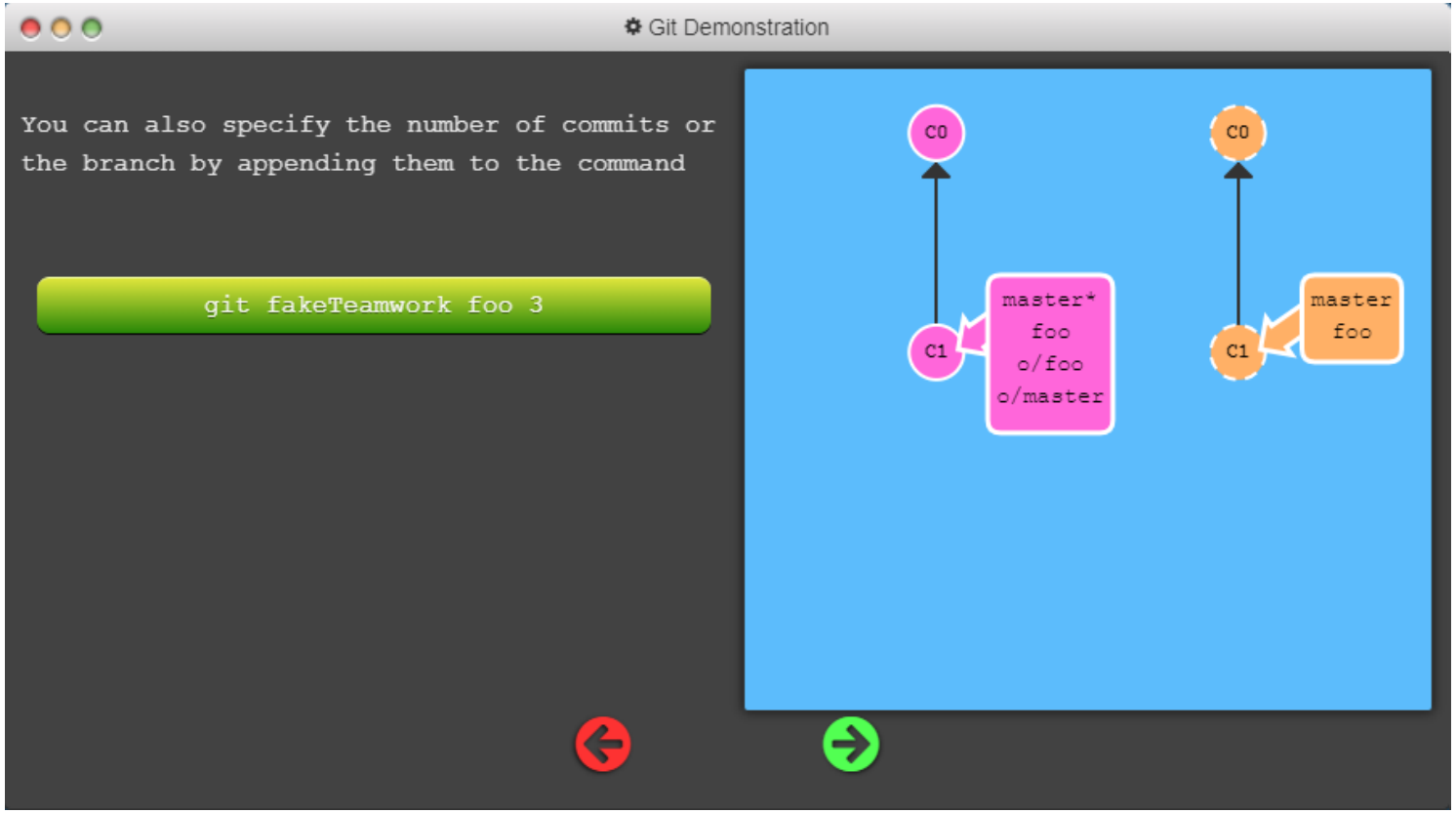
Homework #1.1 – Git Practice

$ git fakeTeamwork (<- Only for this exercise)



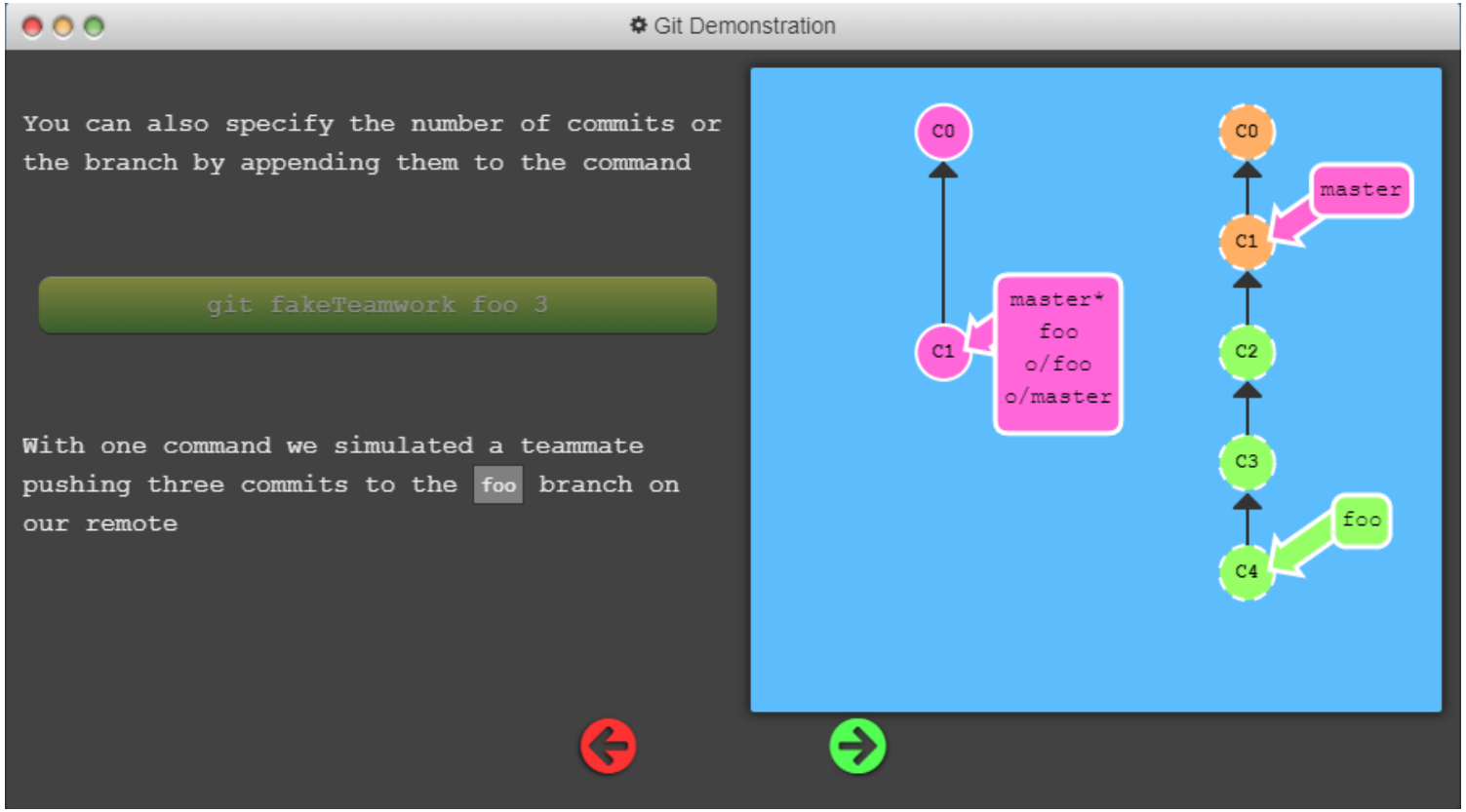
Homework #1.1 – Git Practice

$ git fakeTeamwork foo 3



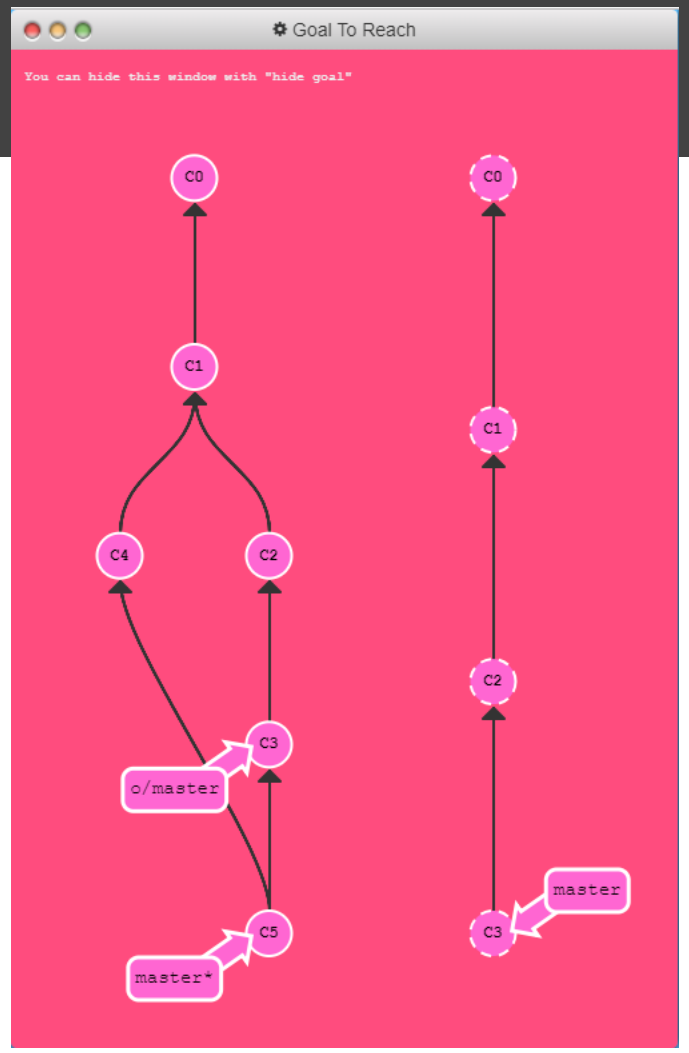
Homework #1.1 – Git Practice

$ git fakeTeamwork foo 3



The upcoming levels are going to be pretty difficult, so we're asking more of you for this level.

Go ahead and make a remote (with `git clone`), fake some changes on that remote, commit yourself, and then pull down those changes. It's like a few lessons in one!



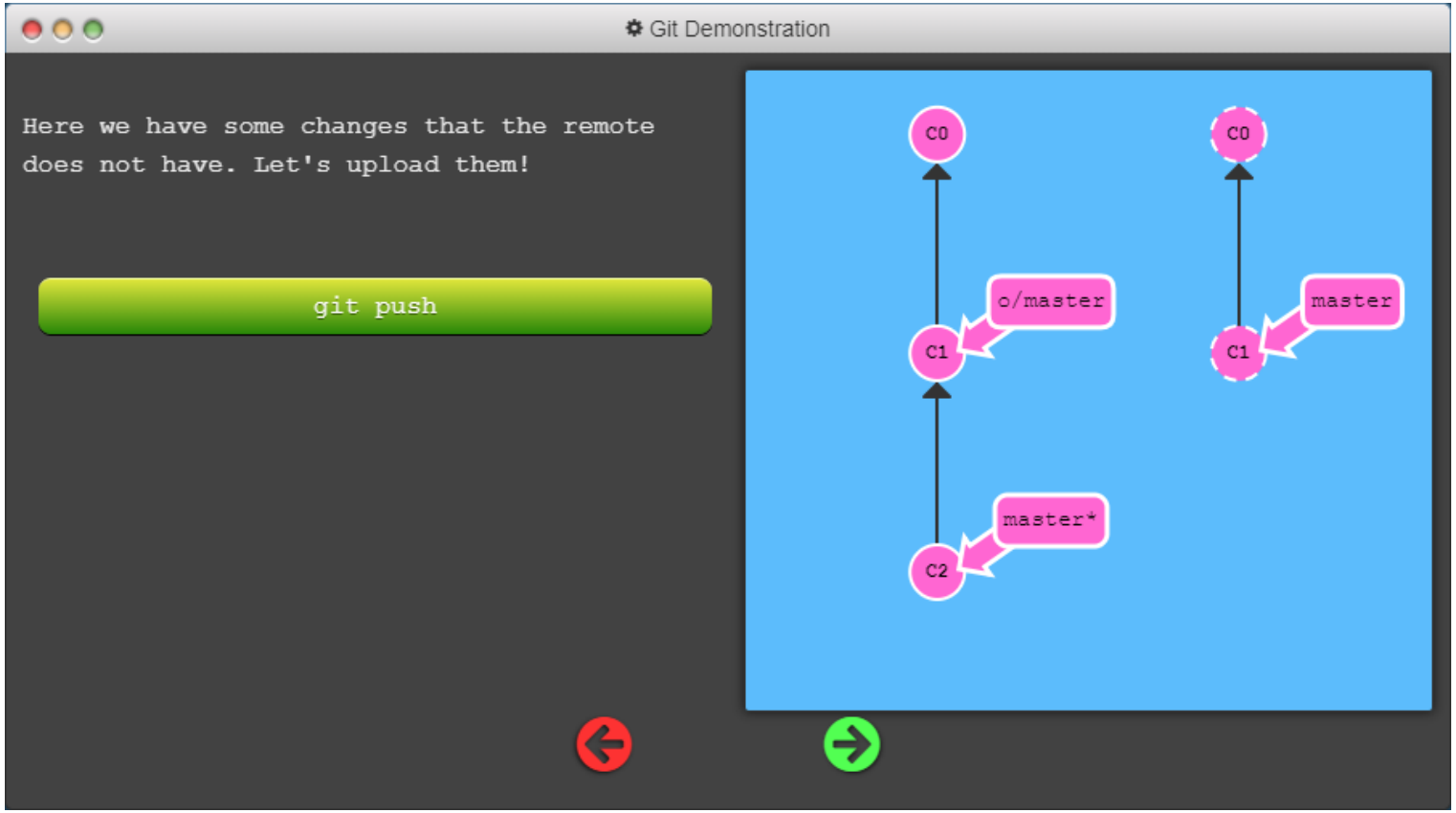
Homework #1.1 – Git Practice

- Git Push
  - *git push*: Share your work from your local repository to the remote repository
    - The opposite concept of git pull
  - Others can see your changes with "Git push"
    - You can show your work to others

    *Note that git push settings can differ. For this exercise, upstream is used.

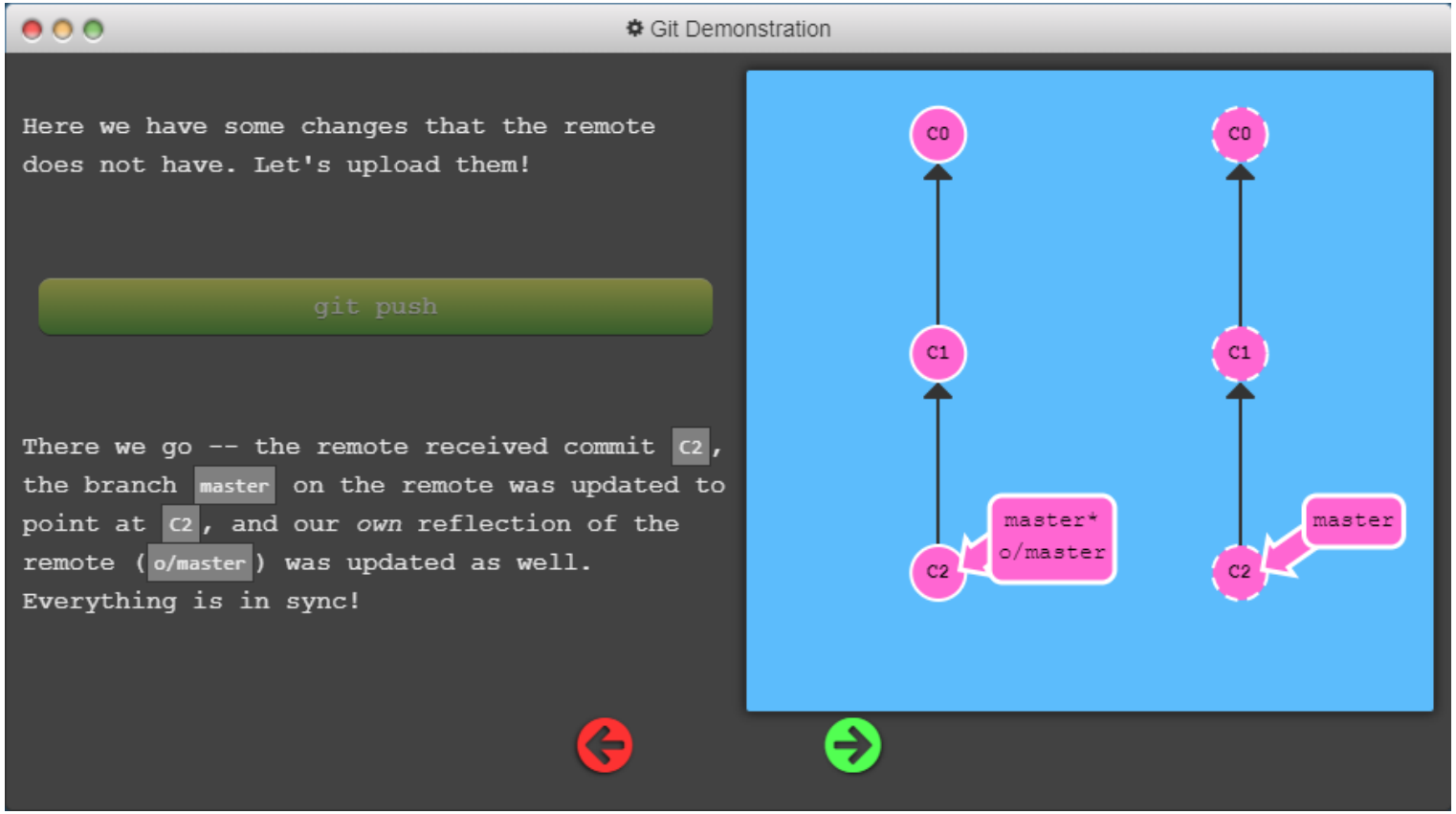
Homework #1.1 – Git Practice

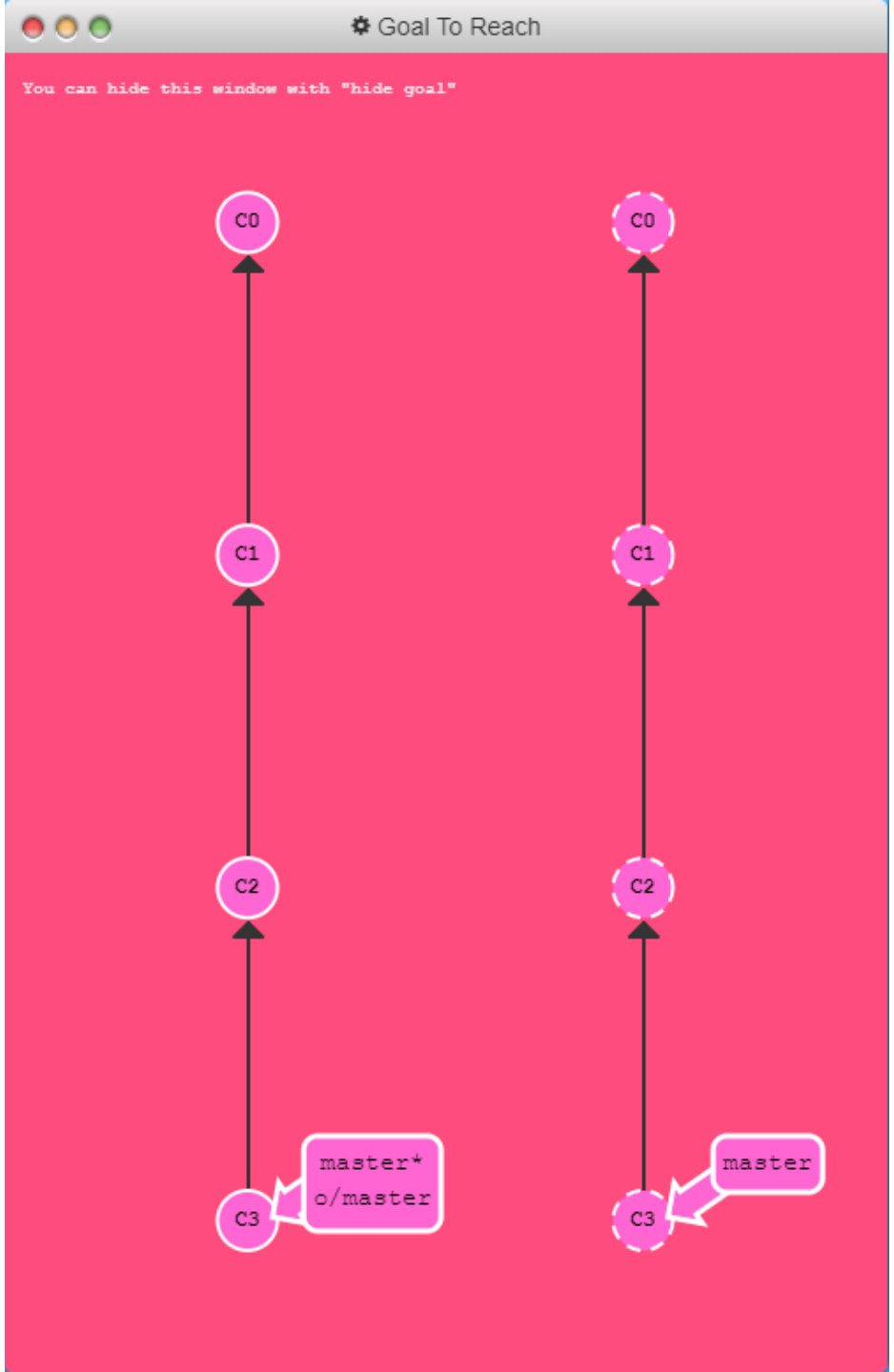
$ git push



Homework #1.1 – Git Practice

$ git push



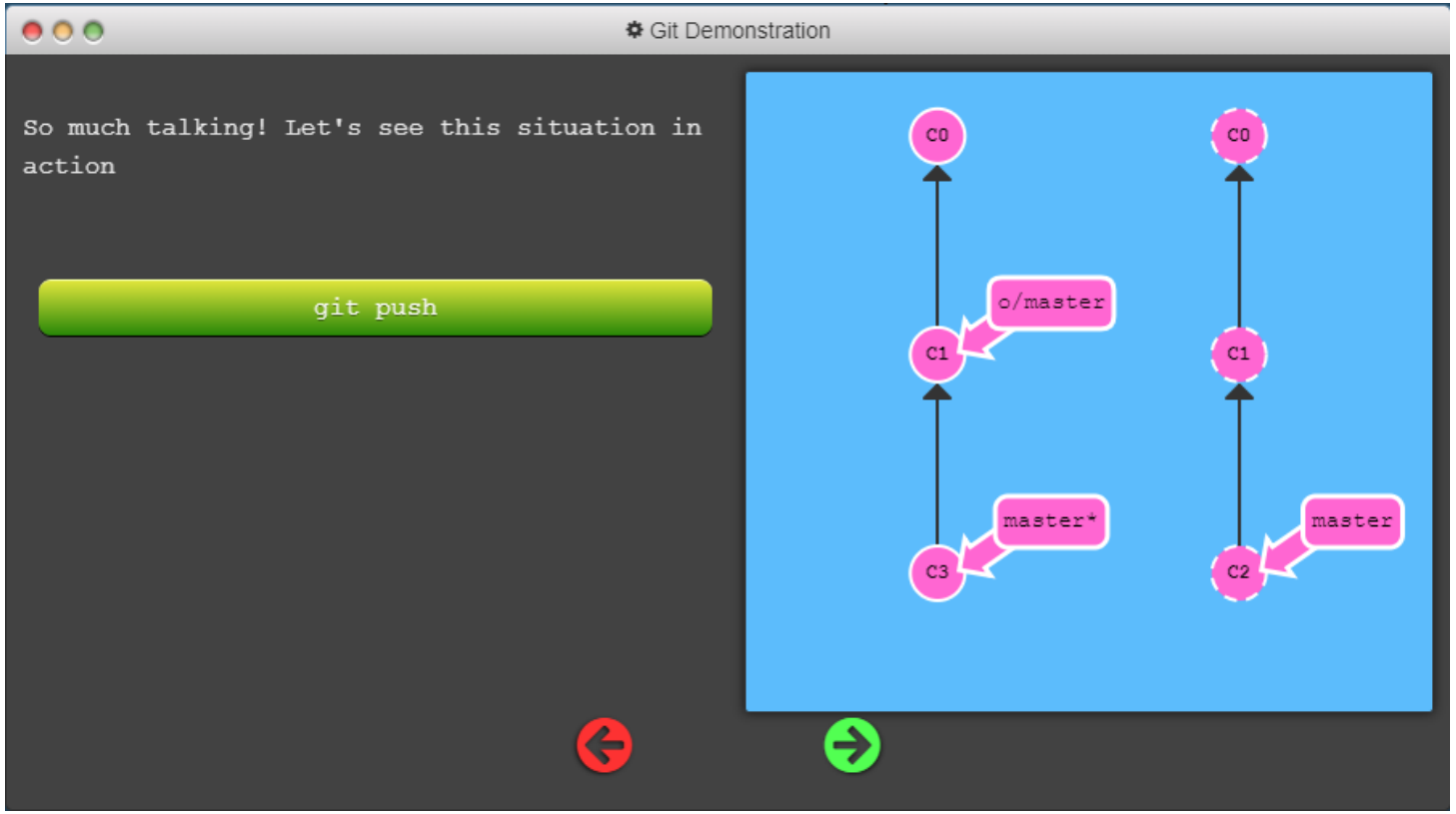

Goal To Reach
You can hide this window with "hide goal"
C0
C1
C2
C3
master*
o/master
C0
C1
C2
C3
master

Homework #1.1 – Git Practice

- Diverged History
  - While I'm pulling, making changes, and pushing, others may have updated the remote repository
    - The version I had worked with is outdated
  - It makes "Push" complicated
    - You need to sync your local repository with the last commit of the remote repository

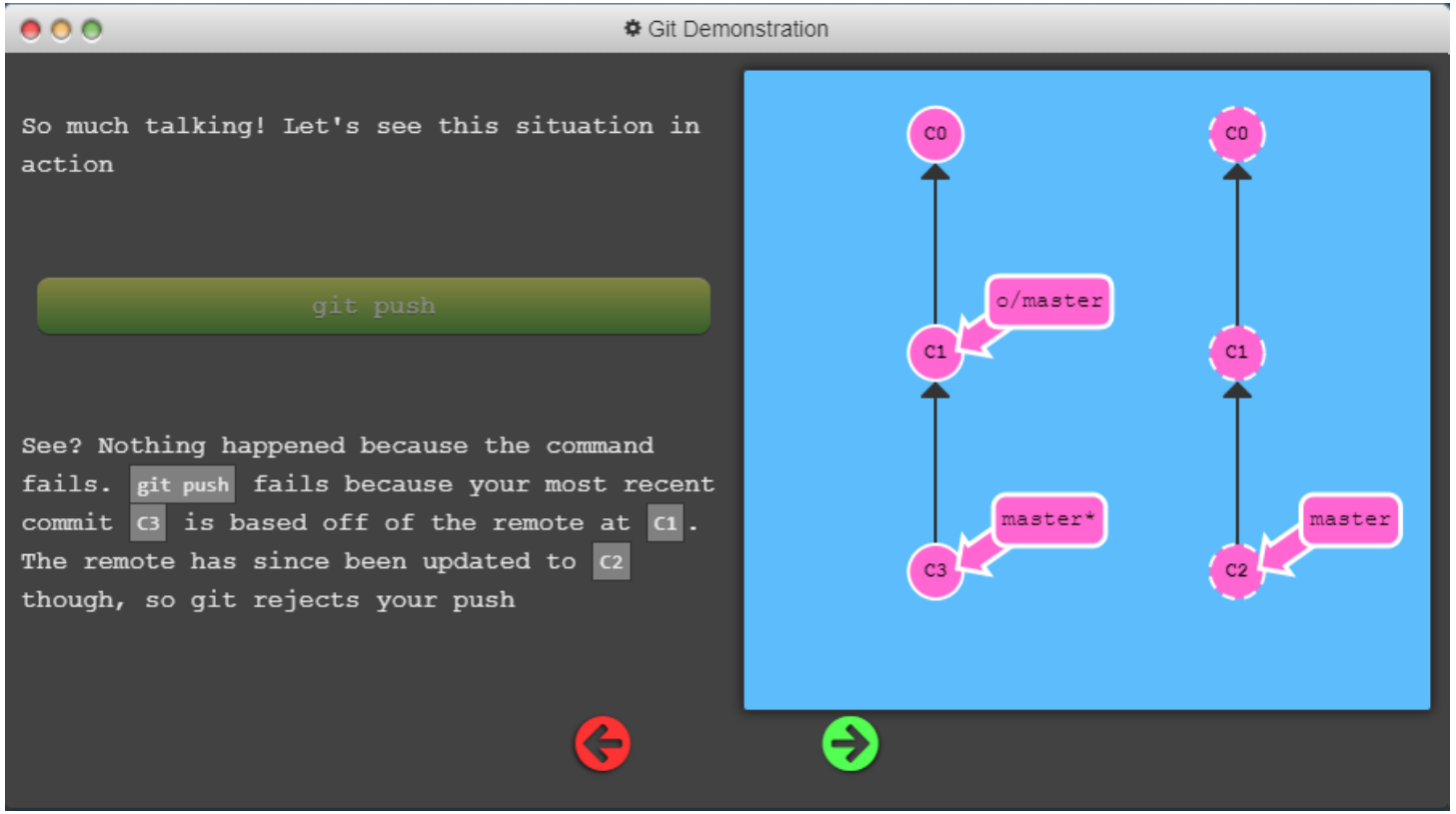
Homework #1.1 – Git Practice

$ git push



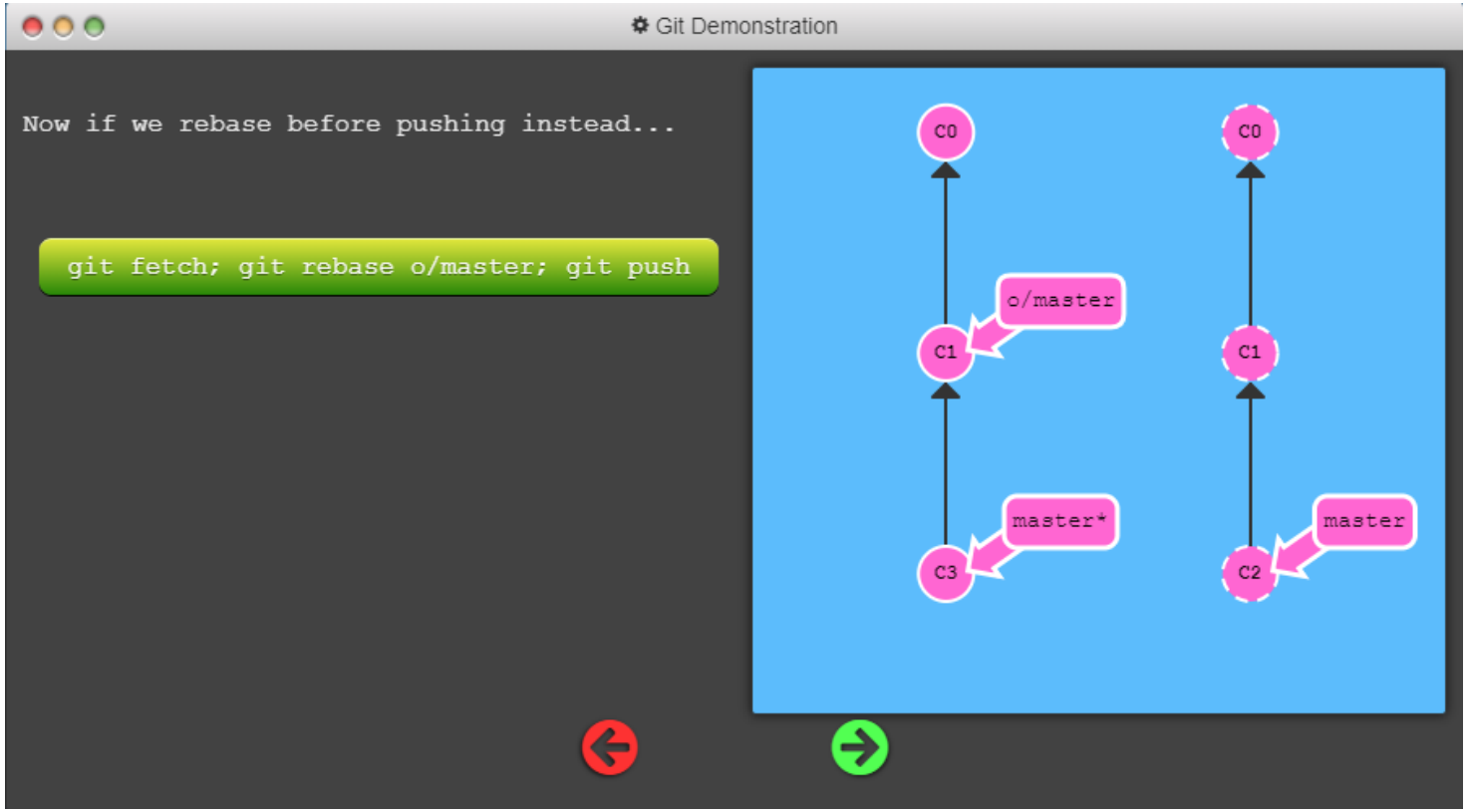
Homework #1.1 – Git Practice

$ git push



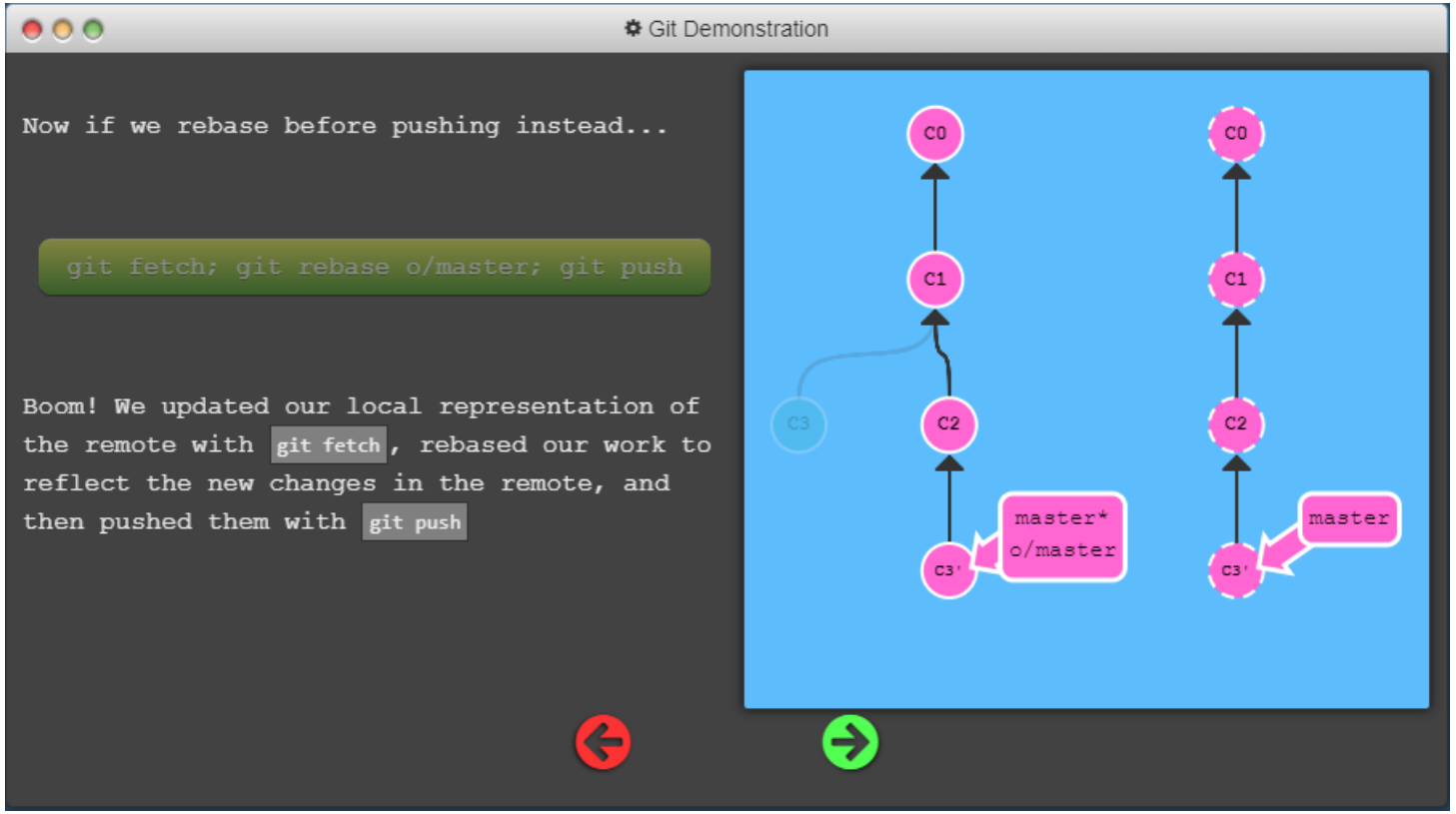
Homework #1.1 – Git Practice

$ git fetch; git rebase o/master; git push



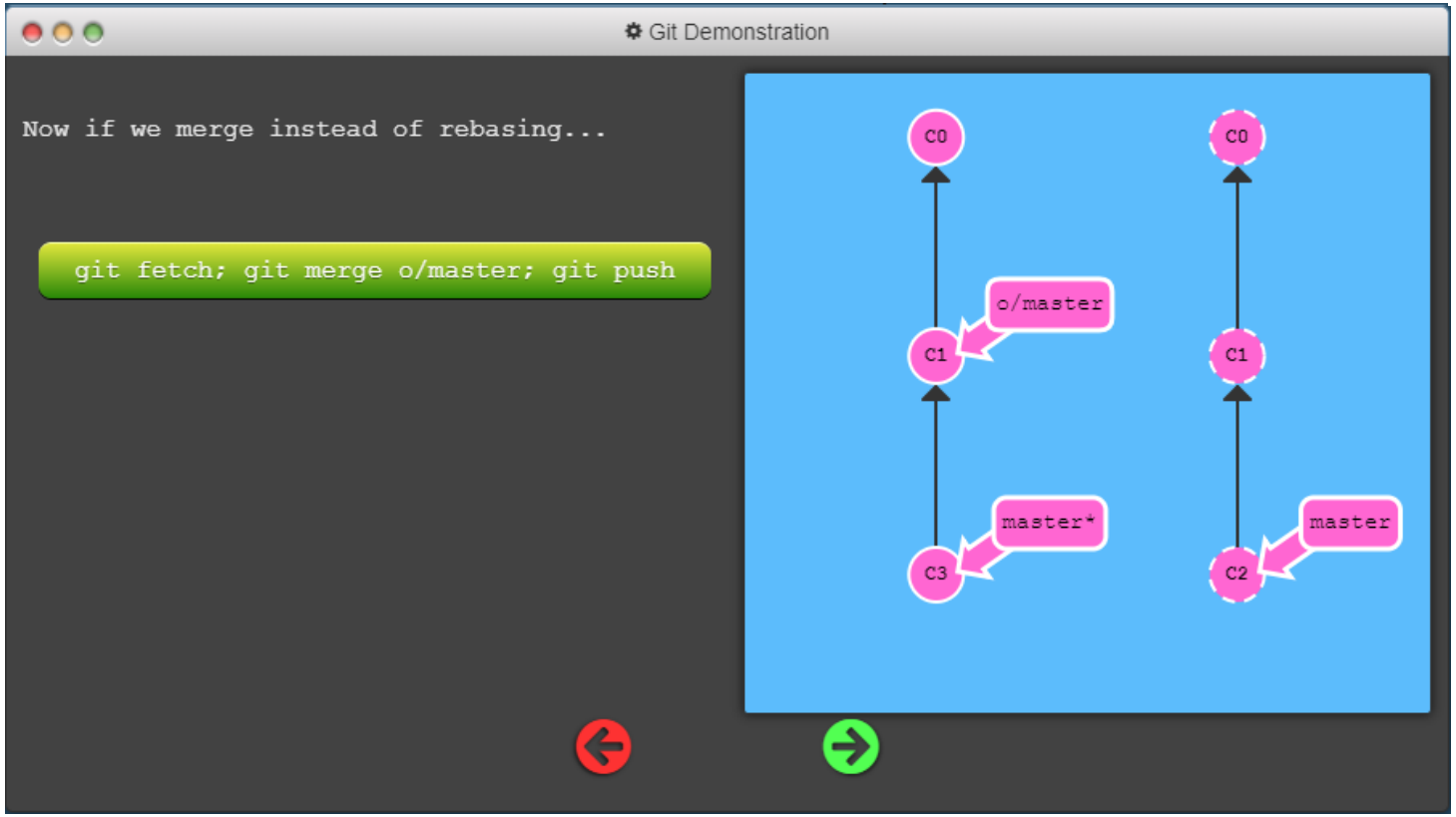
Homework #1.1 – Git Practice

$ git fetch; git rebase o/master; git push



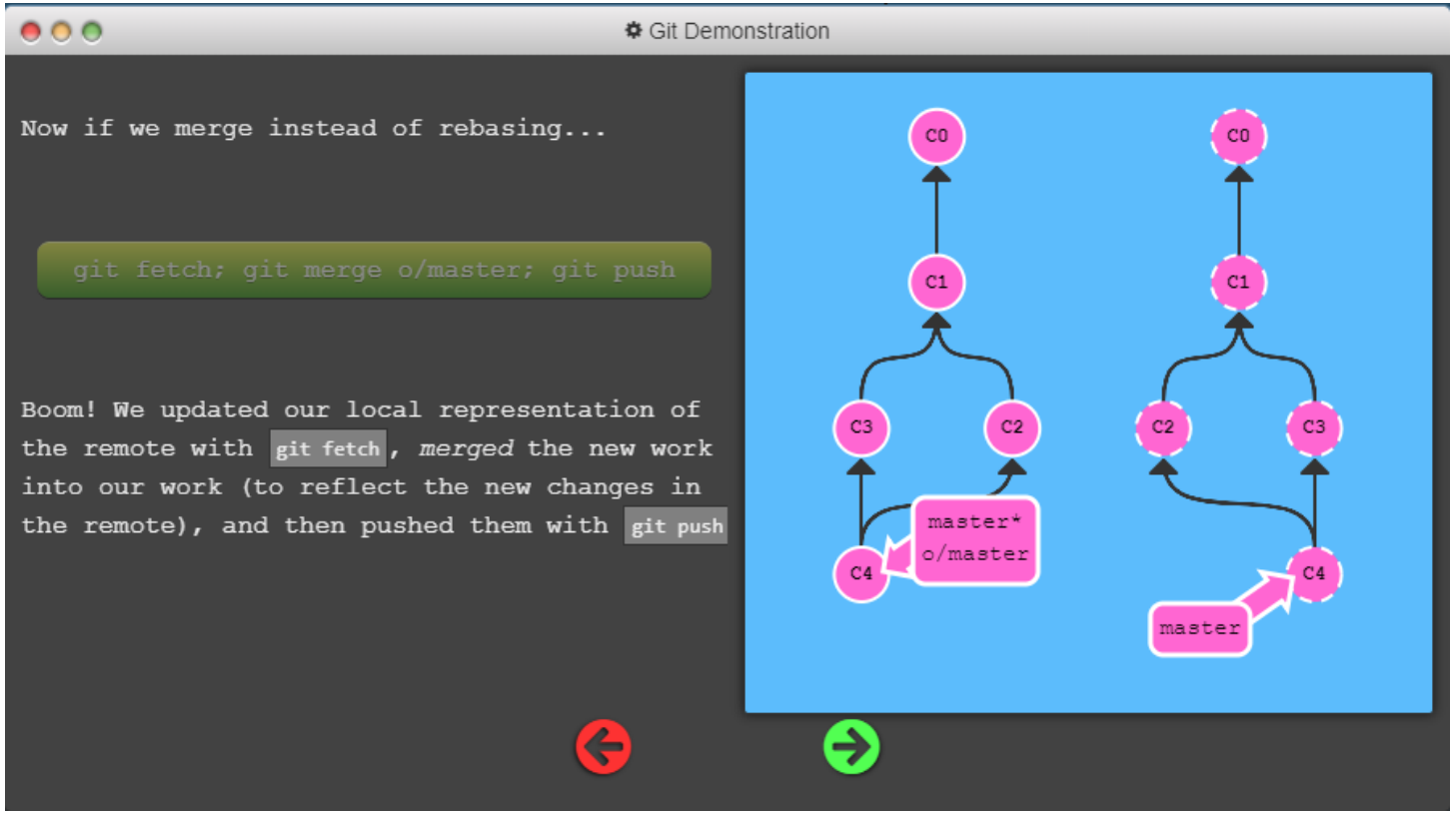
Homework #1.1 – Git Practice

$ git fetch; git merge o/master; git push



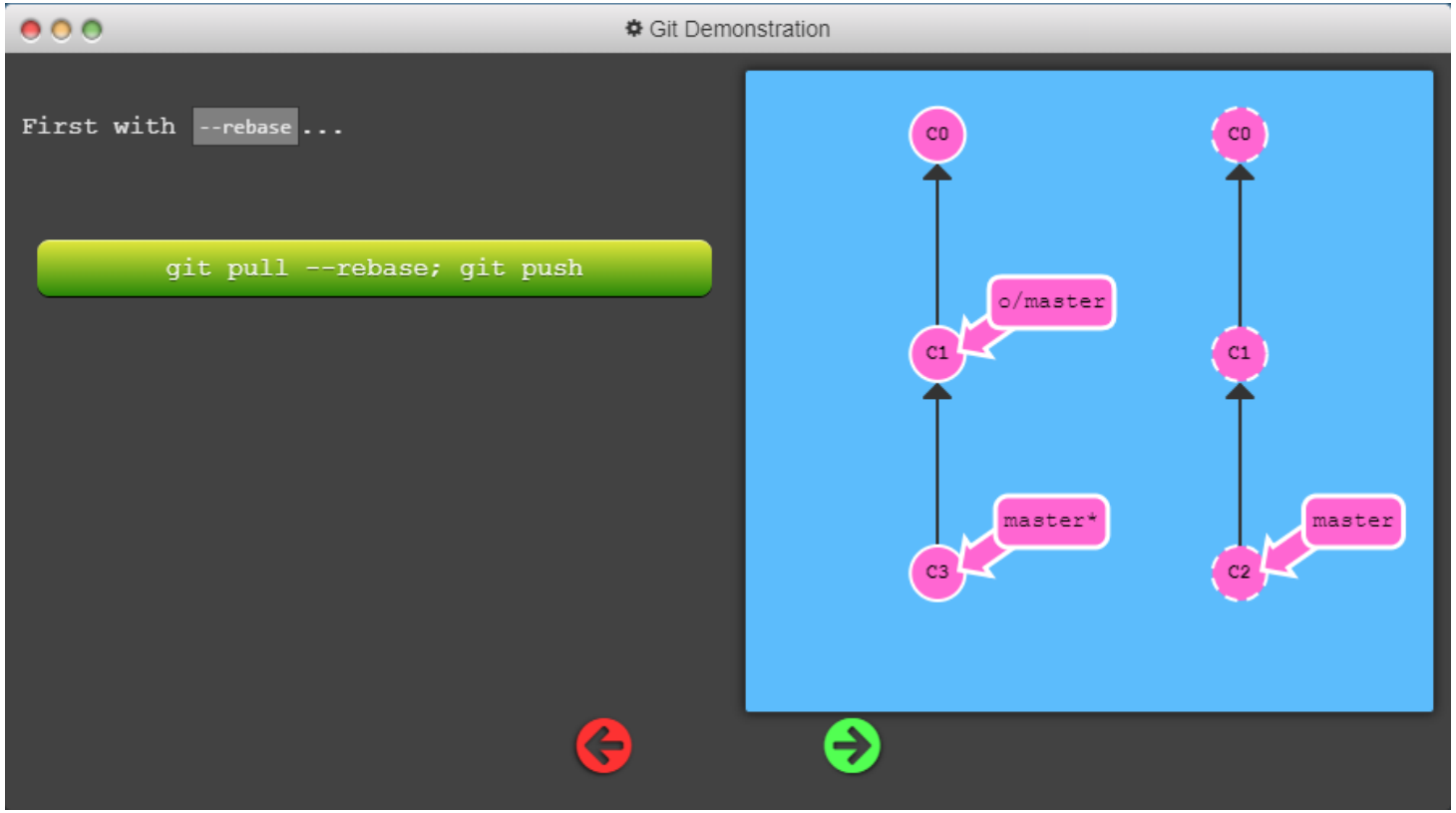
Homework #1.1 – Git Practice

$ git fetch; git merge o/master; git push



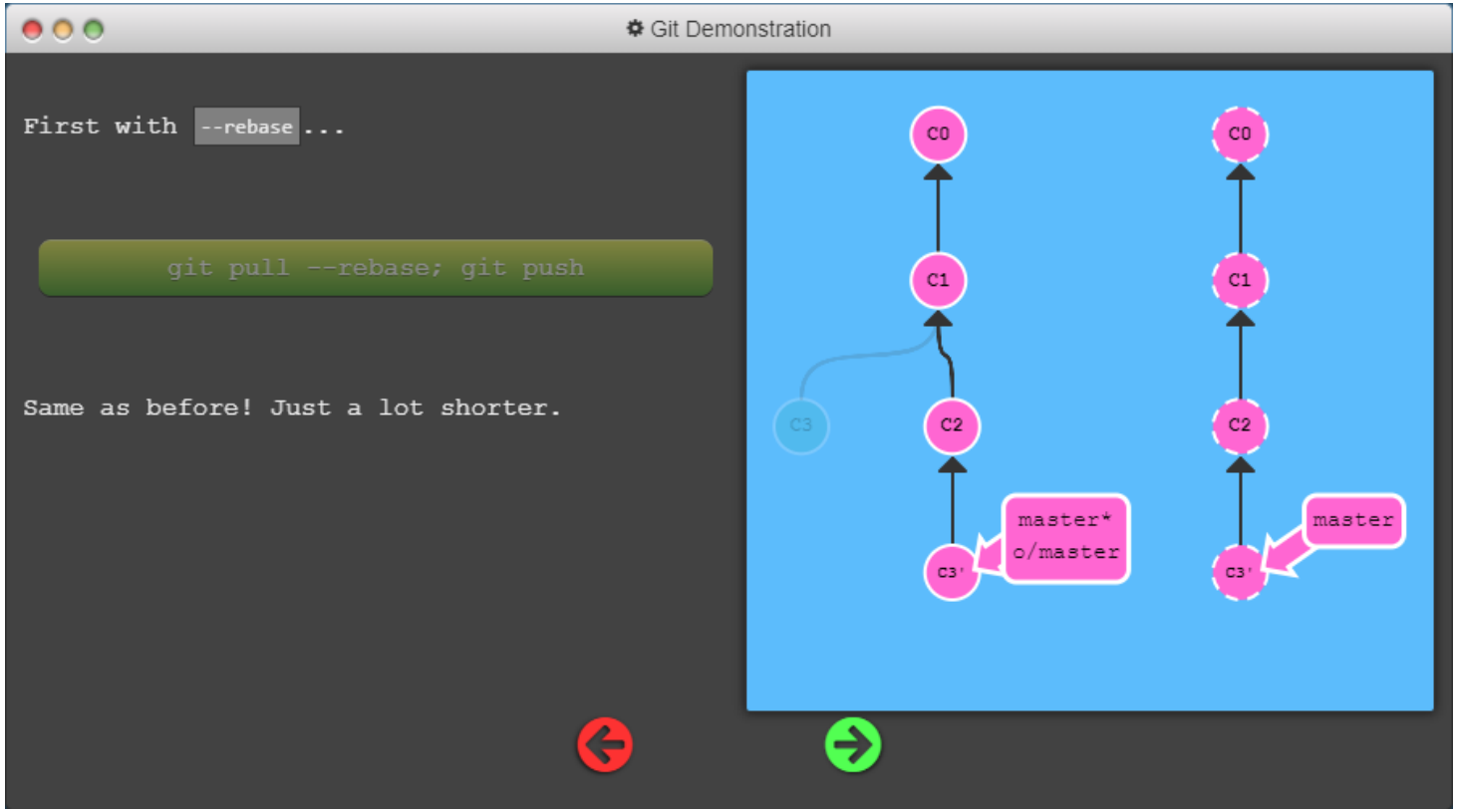
Homework #1.1 – Git Practice

$ git pull –rebase; git push



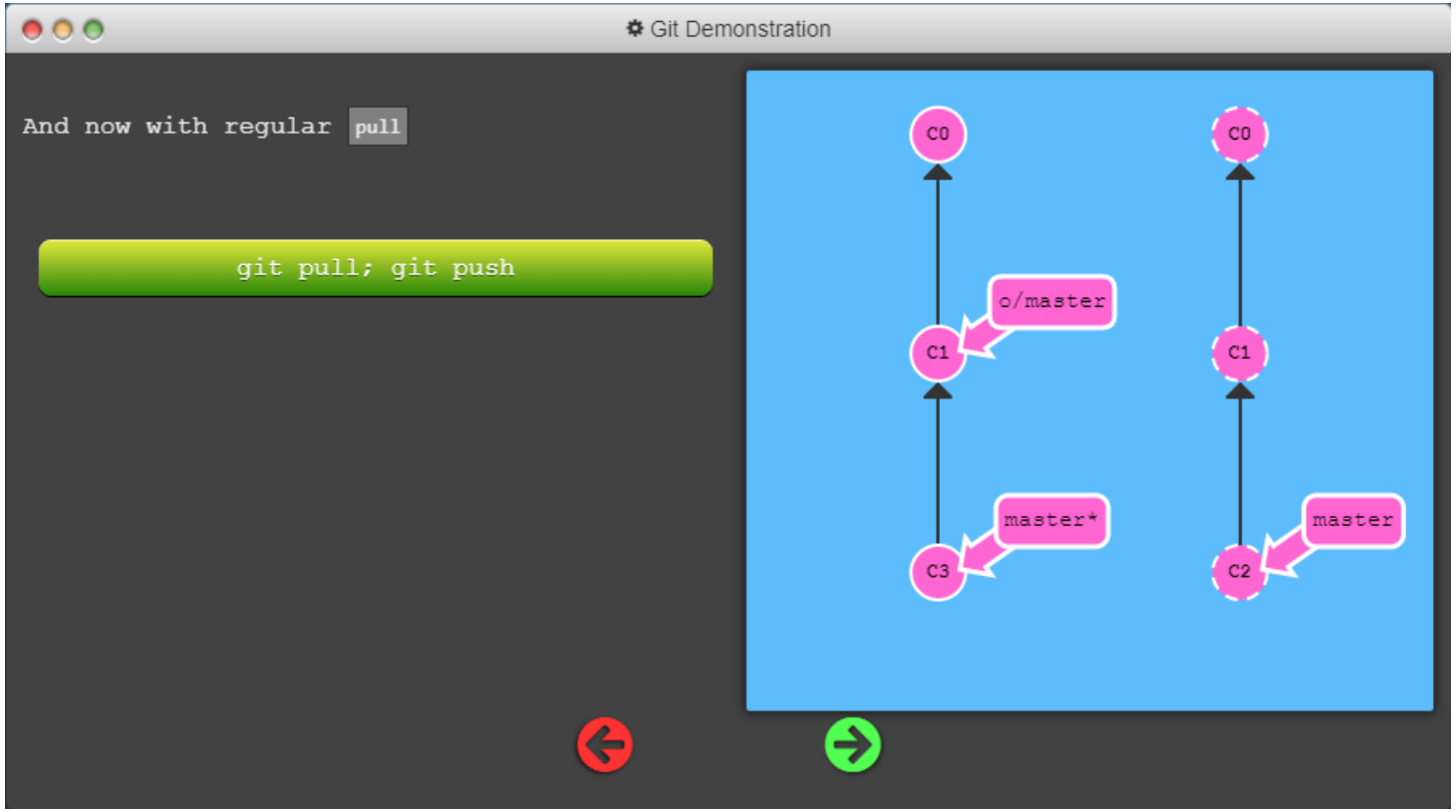
Homework #1.1 – Git Practice

$ git pull –rebase; git push



Homework #1.1 – Git Practice

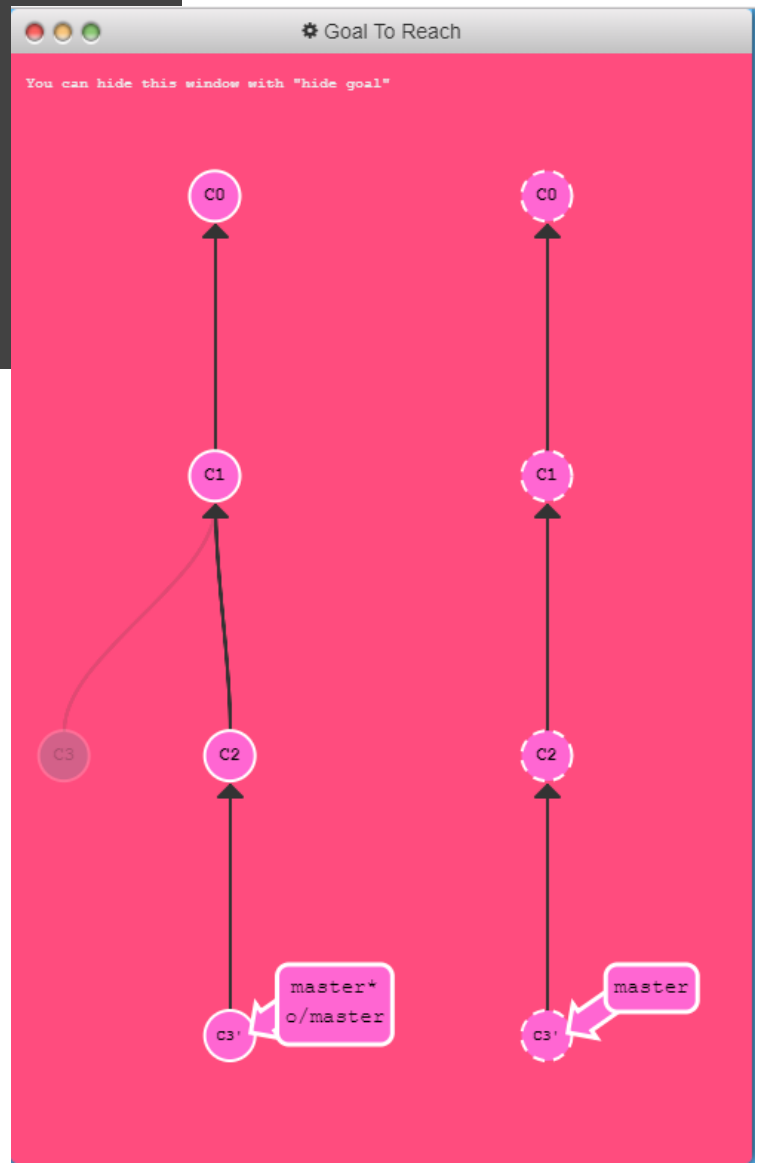
$ git pull; git push



This workflow of fetching, rebase/merging, and pushing is quite common. In future lessons we will examine more complicated versions of these workflows, but for now let's try this out.

In order to solve this level, take the following steps:

- Clone your repo
- Fake some teamwork (1 commit)
- Commit some work yourself (1 commit)
- Publish your work via *rebasing*



Submit another screenshot shows that you have completed the first **8 "remote" exercises** to the Cyber Campus.
Please include your name and Student ID along with the date when taking the screen shot.