

# GIT: From Beginner To Fearless

## GIT Real-Life Activity 03:

Remove a developer's commits from master and retain commits after the developer commits; fix your team members' broken branches

Brian Gorman, Author/Instructor/Trainer

©2019 - MajorGuidanceSolutions

# Introduction

---

Everything is going along smoothly for your team. Multiple developers are working away and getting stuff done. Everyone is playing nicely together, creating feature branches, merging via Pull Requests, and, in general, following the rules.

However, stuff can still go wrong, even when everyone is playing nicely. Consider the situation where developer one created a feature with a couple of commits and the commits were approved and merged. Then developer two performed a rebase and also submitted a pull request, which was merged.

Later, another party (client, manager, ba, etc) comes back to your team and says “I have bad news. The feature that was implemented by Developer one is no longer approved, and we need to remove that functionality from our system. Can you do that?”

Of course there are multiple ways to do this, some safely, and some not-so-safely. If we wanted to play this safely, we could revert all of the commits from the two developers, and then re-apply the feature commits. We could just try to revert the two commits and see what happens. Either way, a public tree with reverted commits should be safe.

For the purpose of learning, however, we’re not going to play it safely this time. We want to do some surgery on our tree, so that we can make it as if these commits never happened, and keep the two commits that we want in our repository.

Before we do this, we need to make it clear to everyone -> No branches are to be created from any of the commits that exist from the PRs merged for developer 1 and developer 2’s feature commits. Additionally, if any branches exist from any of these commits, the branches are going to need to be reworked after the repository is updated. We’ll dive in and see how all of this will work (it’s also going to be an exercise on how to respond to someone mucking up public history that breaks your personal repository tree).

The steps to complete this operation are:

- 1) Make a backup of master [for our own peace of mind – just in case]
- 2) Get everything local, checkout a branch from master that contains all of the commits (good and bad)
- 3) Reset local master to the last good commit before the bad
- 4) Pick the commits to keep
- 5) Once master is correct, force push it to remote [resets and keeps the commits]
- 6) Get other team-members’ branches to work again
- 7) Cleanup by removing old branches.

Let’s get started.

# GFBTF: Git Remove commits from history Activity

## Step 1: Get setup for the activity

- a) Create two commits as developer one, commit and create a PR.  
Developer one is going to be the first one to commit a couple of changes. These are going to be simple changes for this activity. Of course in real-life it will be much more involved. You can make commits on any file, just make sure it is obvious to you where each commit is. For now, I would avoid conflicts but if you want to practice with conflicts, feel free to do so.

```
[git checkout -b activity-three-dev-1-feature]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Defa
(master)
$ git checkout -b activity-three-dev-1-feature
Switched to a new branch 'activity-three-dev-1-feature'
```

Perform two commits that are easy enough to validate are missing later  
[make an obvious change]

```
<li>change to move the master branch forward during
</li>Scenario #3 - Dev 1 - Commit #1</li>
</ol> Brian Gorman, 2 years ago • Added OL for det
```

[commit the change]

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Defa
(activity-three-dev-1-feature)
$ code details.html

blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Defa
(activity-three-dev-1-feature)
$ git add .

blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Defa
(activity-three-dev-1-feature)
$ git commit -m "dev 1 commit 1 - scenario 3"
On branch activity-three-dev-1-feature
nothing to commit, working tree clean
```

[make a second obvious change]

```
<li>Scenario #3 - Dev 1 - Commit #1</li>
<li>Scenario #3 - Dev 1 - Commit #2</li>
</ol>
```

### Notes

[commit the change]

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_workingDirectory
(activity-three-dev-1-feature)
$ git add .

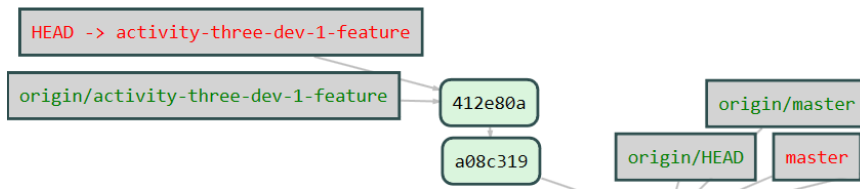
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_workingDirectory
(activity-three-dev-1-feature)
$ git commit -m "dev 1 commit 2 - scenario 3"
[activity-three-dev-1-feature f24ef60] dev 1 commit 2 - scenario 3
1 file changed, 3 insertions(+), 1 deletion(-)
```

Push to remote, and create a PR. Do not merge yet.

[git push origin -u activity-three-dev-1-feature]


```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_workingDirectory
(activity-three-dev-1-feature)
$ git push origin -u activity-three-dev-1-feature
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 353 bytes | 353.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'activity-three-dev-1-feature' on GitHub:
remote:   https://github.com/blgorman/DefaultWeb_workingDirectory/pull/new/activity-three-dev-1-feature
remote:
To https://github.com/blgorman/DefaultWeb_workingDirectory.git
 * [new branch]      activity-three-dev-1-feature -> activity-three-dev-1-feature
Branch 'activity-three-dev-1-feature' set up to track remote branch 'activity-three-dev-1-feature' from 'origin'.
```

Current status:




Create the PR but don't merge it yet:


## Activity three dev 1 feature #5



 **Open** blgorman wants to merge 2 commits into `master` from `activity-three-dev-1-feature`

Conversation 0 Commits 2 Checks 0 Files changed 1



 blgorman commented now Owner + @ ...


First dev feature commits

 blgorman added some commits 5 minutes ago

-  dev 1 commit 1 - scenario 3 a08c319
-  dev 1 commit 2 - scenario 3 412e80a

Add more commits by pushing to the `activity-three-dev-1-feature` branch on blgorman/DefaultWeb\_WorkingDirectory.

  This branch has no conflicts with the base branch  
Merging can be performed automatically.

 Merge pull request You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

- b) Create two commits as developer two, commit and merge via PR.  
Checkout master, then checkout dev-2 feature branch. This will simulate another developer working by getting changes, then rebasing on top of dev 1 changes after merge, then getting changes merged into master.

```
[git checkout master]
[git checkout -b activity-three-dev-2-feature]

blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Def
(activity-three-dev-1-feature)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Def
(master)
$ git checkout -b activity-three-dev-2-feature
Switched to a new branch 'activity-three-dev-2-feature'

blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Def
(activity-three-dev-2-feature)
```

Make a couple of obvious, non-conflicting changes, and commit each change.

```
b1gor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_Worki
(activity-three-dev-2-feature)
$ git add .

b1gor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_Worki
(activity-three-dev-2-feature)
$ git commit -m "Activity three dev 2 change 1"
[activity-three-dev-2-feature 2aa7864] Activity three dev 2 change 1
1 file changed, 1 insertion(+), 1 deletion(-)

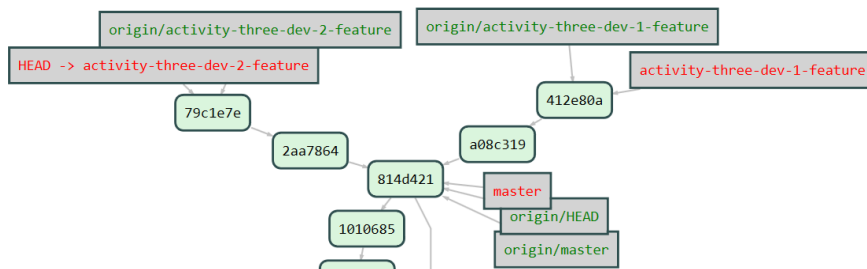
b1gor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_Worki
(activity-three-dev-2-feature)
$ git add .

b1gor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_Worki
(activity-three-dev-2-feature)
$ git commit -m "Activity three dev 2 change 2"
[activity-three-dev-2-feature 79c1e7e] Activity three dev 2 change 2
1 file changed, 1 insertion(+), 1 deletion(-)
```

Push to remote

```
[git push origin -u activity-three-dev-2-feature]
```

```
b1gor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_Worki
(activity-three-dev-2-feature)
$ git push origin -u activity-three-dev-2-feature
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
```



Create the pull request for developer 2's changes, but do not merge it

[Code](#)
[Pull requests 2](#)
[Projects 0](#)
[Wiki](#)

Filters

☐ 2 Open ☒ 4 Closed Author ▾

- ☐ **Activity three dev 2 feature**  
 #6 opened now by blgorman
- ☐ **Activity three dev 1 feature**  
 #5 opened 14 minutes ago by blgorman

c) Create a branch from current master (before dev 1 or dev 2's commits merged for developer #3.

[Code](#)
[Pull requests 2](#)
[Projects 0](#)

This is the working directory of Default Web

[Manage topics](#)

46 commits
 3 branches

Branch: master ▾
 [New pull request](#)

Switch branches/tags

Branches
 Tags

Create branch: activity-three-dev-3-feature from 'master'

CSS
 Added back in the l

d) Merge developer 1's changes, and delete branch:

[Code](#)
[Pull requests 1](#)
[Projects 0](#)
[Wiki](#)
[Security](#)
[Insights](#)
[Settings](#)

This is the working directory of Default Web [Edit](#)

[Manage topics](#)

49 commits
 3 branches
 0 releases
 2 contributors
 MIT

Branch: master ▾
 [New pull request](#)
[Create new file](#)
[Upload files](#)
[Find file](#)
[Clone or download ▾](#)

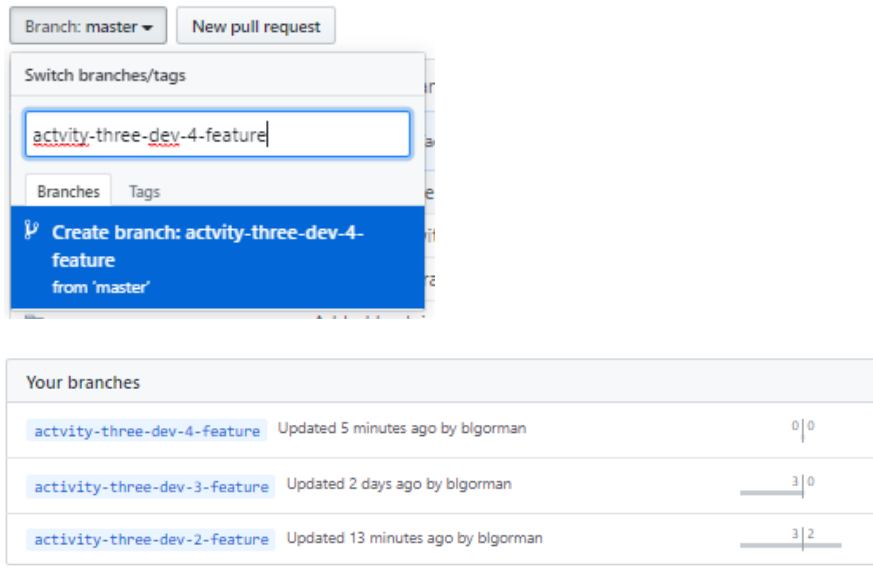
This branch is 17 commits ahead of majorguidancesolutions:master.

[Pull request](#)
[Compare](#)

blgorman Merge pull request #5 from blgorman/activity-three-dev-1-feature
 Latest commit 95be31c 1 minute ago

e) Create a branch from new public commits after dev1 merged as developer #4

Create a branch from merged public commits that will later be removed. This will create a very large problem once public history is rewritten for developer 4.



f) Rebase master on to dev 2, commit, and push, then commit the PR:

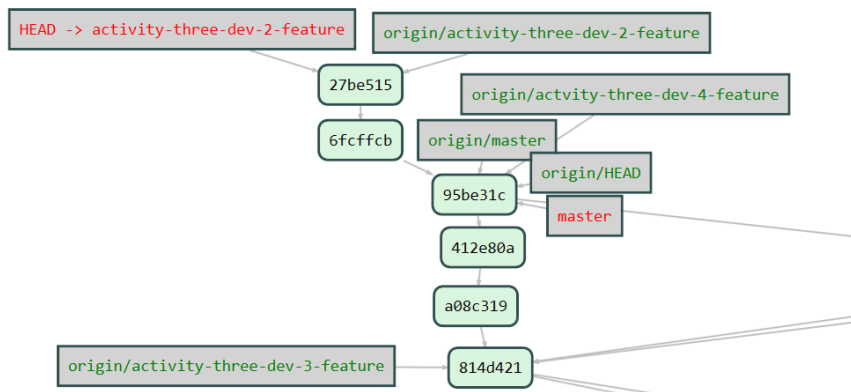
```
[git checkout master]
[git fetch --prune]
[git pull]
[git checkout activity-three-dev-2-feature]
[git rebase origin/master]
[git push --force-with-lease]
[git branch -d activity-three-dev-1-feature]
```

After performing all of those commands, dev 2 should be two commits ahead of master, and dev 1 branch should not show. Dev 3 points to master before any changes, dev 4 is master after dev 1 changes

```
[git log --oneline]
```

```
27be515 (HEAD -> activity-three-dev-2-feature, origin/activity-three-dev-2-featu
re) Activity three dev 2 change 2
6fcffcb Activity three dev 2 change 1
95be31c (origin/master, origin/activity-three-dev-4-feature, origin/HEAD, master)
Merge pull request #5 from blgorman/activity-three-dev-1-feature
412e80a dev 1 commit 2 - scenario 3
a08c319 dev 1 commit 1 - scenario 3
814d421 (origin/activity-three-dev-3-feature) Merge pull request #4 from blgorma
n/pick-and-reset-2
```





- g) Merge dev 2 at remote, so that all changes are in public history;  
delete the branch

This is the working directory of Default Web

[Manage topics](#)

52 commits
3 branches
0 releases

Branch: master
New pull request

This branch is 20 commits ahead of majorguidancesolutions:master.

blgorman Merge pull request #6 from blgorman/activity-three-dev-2-feature ...

Default branch

master Updated 1 minute ago by blgorman Default

Your branches

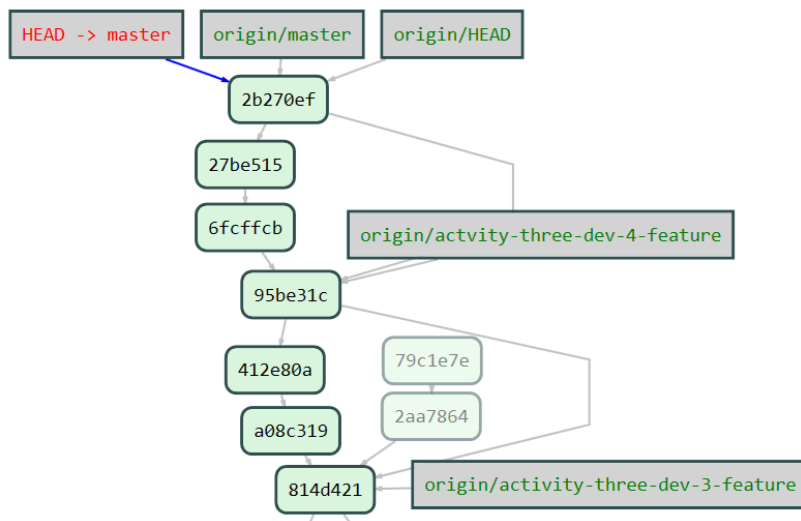
activity-three-dev-4-feature Updated 14 minutes ago by blgorman 3 | 0

activity-three-dev-3-feature Updated 2 days ago by blgorman 6 | 0

- h) Get local up to speed, delete dev 2 branch locally
- ```

[git checkout master]
[git fetch --prune]
[git pull]
[git branch -d activity-three-dev-2-feature]

```



```

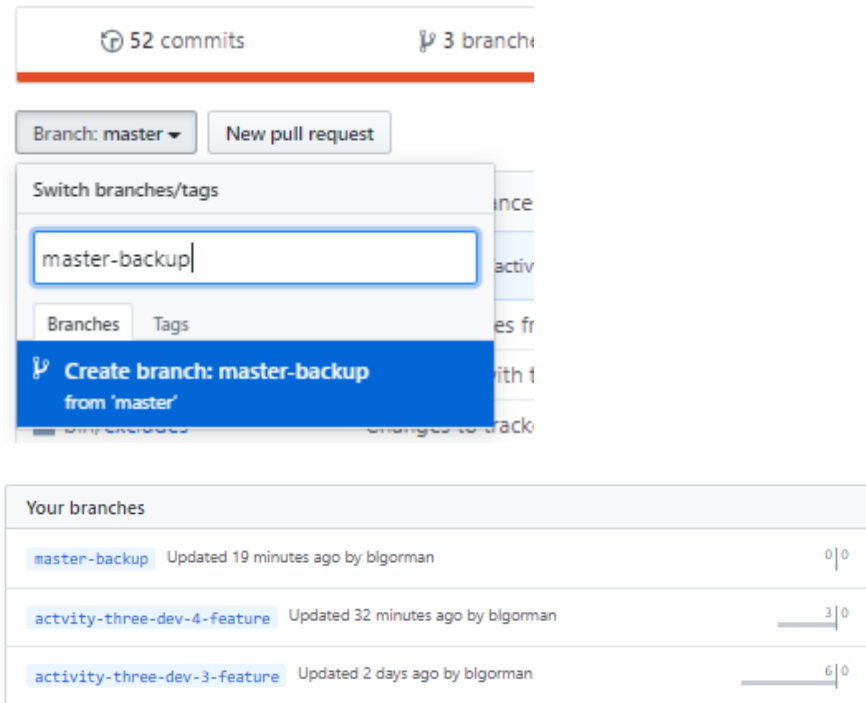
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/default
(master)
$ git branch -a
* master
remotes/origin/HEAD -> origin/master
remotes/origin/activity-three-dev-3-feature
remotes/origin/activity-three-dev-4-feature
remotes/origin/master
remotes/upstream/master

```

Finally, we are setup for the activity!

## Step 2: Backup Master at remote, checkout master locally and create a branch to store all commits

- a) Create the master backup at GitHub for our own peace of mind:



- b) Checkout master locally, and make a fix backup copy to keep all commits. Make notes of all commit ids that are important. We'll use the backup branch to save all commits we need.

```
[git checkout -b activity-three-fix-backup]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/
(master)
$ git checkout -b activity-three-fix-backup
Switched to a new branch 'activity-three-fix-backup'
```

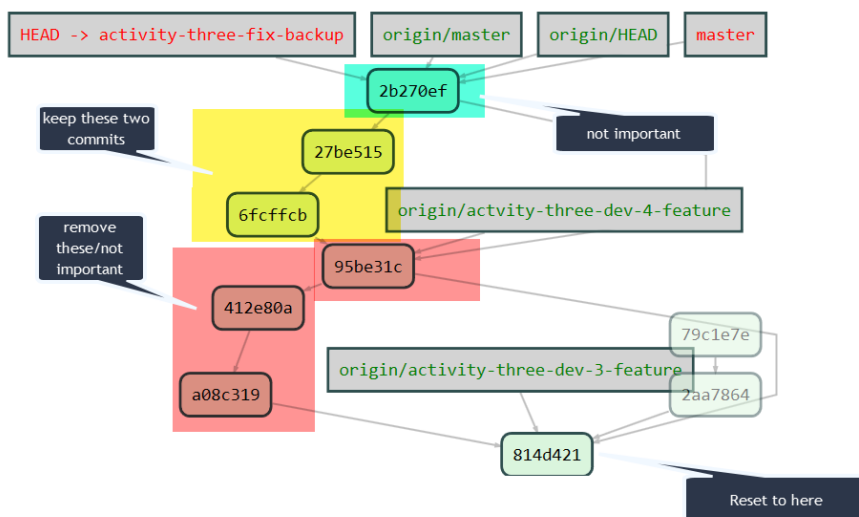
## Step 3: Return to master, and reset hard to original commit before dev 1 and dev 2 changes were merged

- a) Identify the commits to keep and the commits to remove  
[git checkout master]  
[git log --oneline]

```

blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_W
(master)
$ git log --oneline
2b270ef (HEAD -> master, origin/master, origin/HEAD, activity-thr
Merge pull request #6 from blgorman/activity-three-dev-2-feature
27be515 Activity three dev 2 change 2
6fcffcb Activity three dev 2 change 1
95be31c (origin/activity-three-dev-4-feature) Merge pull request #
/activity-three-dev-1-feature
412e80a dev 1 commit 2 - scenario 3
a08c319 dev 1 commit 1 - scenario 3
814d421 (origin/activity-three-dev-3-feature) Merge pull request
n/pick-and-reset-2

```



- b) Go to GitHub and move dev 3 and dev 4 branches ahead one commit  
 Let's assume they have been working and have some public history changes.  
 To make it more difficult, create multiple commits on each branch.

Your recently pushed branches:

- activity-three-dev-3-feature (1 minute ago) [Compare & pull request](#)
- activity-three-dev-4-feature (less than a minute ago) [Compare & pull request](#)

Branch: activity-three-... [New pull request](#) [Create new file](#) [Upload files](#) [Find File](#) [Clone or download](#)

This branch is 18 commits ahead of majorguidancesolutions:master. [Pull request](#) [Compare](#)

blgorman Update About.html Latest commit c2ffb02 now

Your branches

| Branch                       | Updated                           | Commits | Actions                                                 |
|------------------------------|-----------------------------------|---------|---------------------------------------------------------|
| activity-three-dev-4-feature | Updated 1 minute ago by blgorman  | 3   1   | <a href="#">New pull request</a> <a href="#">Delete</a> |
| activity-three-dev-3-feature | Updated 2 minutes ago by blgorman | 6   1   | <a href="#">New pull request</a> <a href="#">Delete</a> |
| master-backup                | Updated 1 hour ago by blgorman    | 0   0   | <a href="#">New pull request</a> <a href="#">Delete</a> |

```

[git fetch --prune]
[git pull]
[git branch -a]

```

```

b1gor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_WorkingDirectory
(master)
$ git branch -a
  activity-three-fix-backup
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/activity-three-dev-3-feature
  remotes/origin/activity-three-dev-4-feature
  remotes/origin/master
  remotes/origin/master-backup
  remotes/upstream/master

```

c) Reset master with a hard reset to 814d421

```
[git reset --hard 814d421]
```

```

b1gor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_WorkingDirectory
(master)
$ git reset --hard 814d421
HEAD is now at 814d421 Merge pull request #4 from b1gorman/pick-and-reset-2

```

Master is now back to where it was before either developer committed changes. Now it gets interesting. We know there are two other developers with changes that are pending. WE can fix things for them, or we can force push and they will be forced to clean up their branches. Either way, we should make sure they know not to do any more changes to their branches until we get things cleaned up.

Additionally, we need to consider – does it matter when the other developers branched? For example, do we want to reset dev 3 to before or after dev 2's original changes? The scenario of course will depend on your real-life situation.

For our purposes, we are going to keep dev 2's changes and get dev 3 and 4 restored to the end of the chain. Things would be more difficult if they had also made changes.

d) Checkout all other developer branches

This way we can fix everything locally. In the real world, it will likely be messier than this. Likely, one or more developers will have to fix things on their own branches.

```
[git checkout activity-three-dev-4-feature]
```

\*I spelled it wrong at GitHub

```
[git checkout activity-three-dev-3-feature]
```

Which shows us we need to keep the commits on branch dev 3 and the commits on branch dev 4. We'll fix all of this as we go, for now, we are in a good place.

e) Get the "good" commits onto master

Again, we're assuming these two commits are important and other features need to be on top of them. Pick each commit onto the master

```
[git checkout master]
[git cherry-pick 6fcffcb]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb
(master)
$ git cherry-pick 6fcffcb
[master 655cef3] Activity three dev 2 change 1
Date: Fri Aug 2 22:34:08 2019 -0500
1 file changed, 1 insertion(+), 1 deletion(-)
```

If you want, you could verify the file affected to make sure you have the details. This could be manual or with a `[git diff <commit-1> <commit-2>]` command

```
[git cherry-pick 27be515]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb
(master)
$ git cherry-pick 27be515
[master 8c6ac17] Activity three dev 2 change 2
Date: Fri Aug 2 22:35:09 2019 -0500
1 file changed, 1 insertion(+), 1 deletion(-)
```

This gets both the commits we want to keep on top of master.

```
[git log --oneline]
```

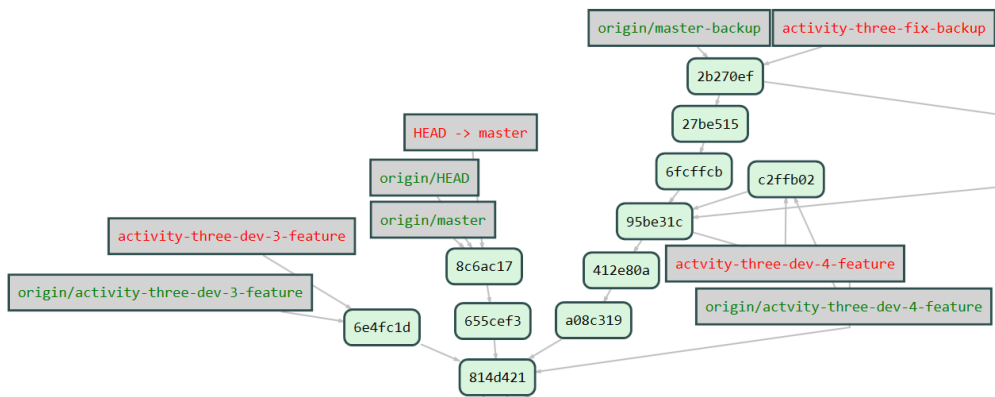
```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb
(master)
$ git log --oneline
8c6ac17 (HEAD -> master) Activity three dev 2 change 2
655cef3 Activity three dev 2 change 1
814d421 Merge pull request #4 from blgorman/pick-and-reset-2
1010685 Making more important changes #4
```

Now that master is where it should be, force-push it to remote so that history is officially rewritten.

```
[git push origin --force]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb
(master)
$ git push origin --force
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
```

This gets master on remote back to where we want it. The dev 1 commits are no longer in history.



## Step 4: Fix developer 3 and developer 4 branches while maintaining their changes so that they may resume work

- a) Fixing developer 3 is going to be easy – because their parent history commit still exists at the public repository

Even better news. Since dev 3 had branched off of the original master and that original master commit is still in our “new” history, developer 3 only needs to do a simple rebase, and their branch is good to go. Likely, developer three would do this on their own machine, but I’ll fix it here:

```
[git checkout activity-three-dev-3-feature]
[git rebase origin/master]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Defau
(master)
$ git checkout activity-three-dev-3-feature
Switched to branch 'activity-three-dev-3-feature'
Your branch is up to date with 'origin/activity-three-dev-
activity-three-dev-3-feature'
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Defau
(activity-three-dev-3-feature)
$ git rebase origin/master
First, rewinding head to replay your work on top of it...
Applying: Update index.html
```

And then I’ll just do a force-push to make sure their branch is correct at origin. They would have done the same thing from their own branch anyway. Also, I’ll use force with lease, in case they didn’t listen and kept developing on their branch.

```
[git push --force-with-lease]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/  
(activity-three-dev-3-feature)  
$ git push --force-with-lease  
Enumerating objects: 5, done.  
Counting objects: 100% (5/5), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (3/3), 343 bytes | 343.00 KiB/s  
Total 3 (delta 2), reused 0 (delta 0)  
remote: Resolving deltas: 100% (2/2), completed with  
To https://github.com/blgorman/DefaultWeb_workingDire  
+ 6e4fc1d...77cefa6 activity-three-dev-3-feature ->  
e (forced update)
```

In the real world, dev 3 would then checkout master and pull latest from master, then do a reset of their branch from origin by running this command [you should not run it in this scenario]

```
[git reset --hard origin/activity-three-dev-3-  
feature]
```



[we aren't going to do anything else with dev 3 branch – it's good to go from here].

#### b) Fix the developer 4 branch

Fixing developer 4's branch is not going to be that easy. This branch has the master in its history, but was based on a branch that no longer exists. A simple rebase won't work, because we'll keep the commits we don't want. Therefore, we need to do this differently.

First, checkout a new branch off local master, and then pick the one commit we need.



```
[git checkout master]
[git checkout -b new-dev-4-branch]
[git cherry-pick c2ffb02]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/default
(activity-three-dev-3-feature)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/default
(master)
$ git checkout -b new-dev-4-branch
Switched to a new branch 'new-dev-4-branch'

blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/default
(new-dev-4-branch)
$ git cherry-pick c2ffb02
[new-dev-4-branch 9a77b9c] Update About.html
Author: Brian Gorman <blgorman@gmail.com>
Date: Fri Aug 2 23:48:07 2019 -0500
1 file changed, 4 insertions(+), 1 deletion(-)
```

Here we could compare to make sure that the commit is as expected

```
[git log --oneline]
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_workingDire
(new-dev-4-branch)
$ git log --oneline
9a77b9c (HEAD -> new-dev-4-branch) Update About.html
8c6ac17 (origin/master, origin/HEAD, master) Activity three dev 2 change 2
```

```
[git diff 9a77b9c 8c6ac17] - new to master
```

```
[git diff c2ffb02 95be31c] - old commit to it's parent
```

And we can then compare to see the bits are the same

#### c) Force push the new dev 4 to original dev 4

We could push dev 4 out to remote and they would be able to just pull that and work from it. Let's go ahead and make their original branch be what they would want and then they can reset their local to match

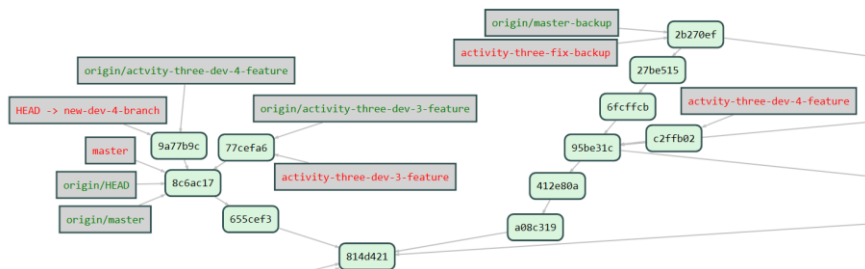
Note that we are going to overwrite their branch, we will use the force with lease option so that we don't lose any commits they may have done if they didn't listen and kept developing and making more commits. Also, it's a good idea before doing this to make sure to talk with the other developer to make sure it's ok to overwrite their branch.

```
[git push origin HEAD:activity-three-dev-4-feature -
force-with-lease]
```

```

b1gor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_workingD
(new-dev-4-branch)
$ git push origin HEAD:activity-three-dev-4-feature --force-with-lease

```



Dev 4 will need to perform a hard reset to get their branch in sync locally.

So now both dev 3 and dev 4 are good to go, and we have the two commits from dev 2 and we have eliminated dev 1's commits. Locally we still have all the references to the older commits, but dev 3 and 4 can go forward, and the rest of the team can also resume making changes off master. If anyone is out of sync, they may need to also run a hard reset against origin.

| Your branches                |                                 |       |                  |  |
|------------------------------|---------------------------------|-------|------------------|--|
| activity-three-dev-4-feature | Updated 1 hour ago by bigorman  | 0   1 | New pull request |  |
| activity-three-dev-3-feature | Updated 1 hour ago by bigorman  | 0   1 | New pull request |  |
| master-backup                | Updated 2 hours ago by bigorman | 2   6 | New pull request |  |

## Step 5: Cleanup

- Delete master-backup at GitHub

We no longer need this.

- Resume development on dev 3 and dev 4, then merge with PRs

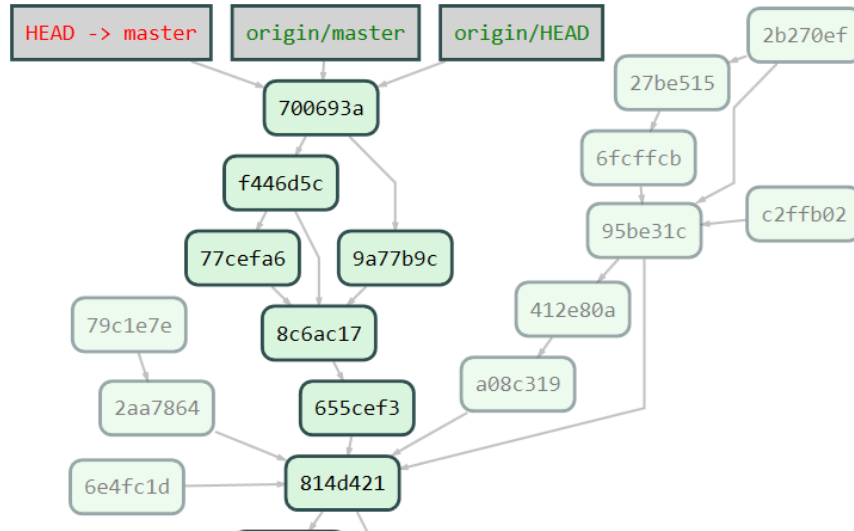
| Overview                     | Yours                           | Active  | Stale | All branches | Search branches...    |
|------------------------------|---------------------------------|---------|-------|--------------|-----------------------|
| Default branch               |                                 |         |       |              |                       |
| master                       | Updated 3 hours ago by bigorman | Default |       |              | Change default branch |
| Your branches                |                                 |         |       |              |                       |
| activity-three-dev-4-feature | Updated 1 hour ago by bigorman  | 0   1   | #5    | Open         |                       |
| activity-three-dev-3-feature | Updated 1 hour ago by bigorman  | 0   1   | #7    | Open         |                       |

Merge dev 3, then merge dev 4 [could rebase, etc]

| Overview             | Yours                        | Active  | Stale | All branches | Search branches...    |
|----------------------|------------------------------|---------|-------|--------------|-----------------------|
| Default branch       |                              |         |       |              |                       |
| master               | Updated now by bigorman      | Default |       |              | Change default branch |
| Your branches        |                              |         |       |              |                       |
| activity-three-de... | Deleted just now by bigorman |         |       |              | Restore               |
| activity-three-d...  | Deleted just now by bigorman |         |       |              | Restore               |

### c) Cleanup local

```
[git checkout master]
[git fetch --prune]
[git pull]
[git branch -D activity-three-fix-backup]
[git branch -D activity-three-dev-4-feature]
[git branch -d activity-three-dev-3-feature]
[git branch -d new-dev-4-branch]
```



### d) [Optional] Expire cached unreachable commits

```
[git reflog expire --expire-unreachable=now --all]
[git gc --prune=now]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/
(master)
$ git reflog expire --expire-unreachable=now --all

blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/
(master)
$ git gc --prune=now
Enumerating objects: 175, done.
Counting objects: 100% (175/175), done.
Delta compression using up to 8 threads
Compressing objects: 100% (86/86), done.
Writing objects: 100% (175/175), done.
Total 175 (delta 86), reused 158 (delta 76)

blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/
[now the local is clear of any orphaned commits]
```

This concludes our Remove commits from history activity.

## Closing Thoughts

In this activity we saw what happens when we need to remove a couple of commits from the public commit history, and had the opportunity to see what it takes to fix up other developers' branches after doing history rewrites.

As we can see, with the right planning and the right information, it is possible to do quite a lot of fantastic things with the GIT history. Here, we completely removed public commits as if they never happened.

While there are always many ways to complete an activity like this, the way I showed you is my preferred action because it's the way that I understand how GIT works the best.

With this activity, combined with other things that we have learned through the course, we are now in a position to be able to work with GIT at some of the most advanced levels possible.

I hope you have found the information and activities useful, and I hope you are now in a position to be able to lead the GIT charge for your team, now, and into the future.

### Notes