```
#Point to another commmit (commit your uncommited changes before doing this)
git checkout [comit_hash]

# Resets index to former commit; replace '56e05fced' with your commit code
git reset 56e05fced

# Moves pointer back to previous HEAD
git reset --soft HEAD@{1}

git commit -m "Revert to 56e05fced"

# Updates working copy to reflect the new commit
git reset --hard
```

```
================================
git branch -r #show remote branches
git branch #show local branches
git branch -v #show local branches and the last commit
git fetch origin #Fetch a branch
git branch -v -a #Check available branches for checkout
git checkout -b my_new_branch #Verify previously the branch you are
git checkout -b my_local_new_branch_name origin/branch_name #Check out the branch you are interested in
================================
Eliminar detached head: Detached HEAD state means that you are on unammed branch. HEAD points directly to commit and is no
reference to some local branch

git reset --hard

================================
Actualizar fork branch

First add the upstream project as a Git remote:

git remote add upstream https://github.com/upstream-username/projectname.git
Then pull (meaning fetch and then merge automatically) the changes from the remote's master branch into your local reposit

git pull upstream master #master means the name of the branch (4.5.6, 4.6.0..)
Now your local repository is in sync with upstream. Finally, push your local repo to your Github fork:

git push origin maste
```

?https://superuser.com/questions/456145/how-can-i-resync-a-fork-from-original/456189#456189?newreg=301978e6c06a41a680396c82fc589f3c
?http://stackoverflow.com/questions/4114095/revert-to-a-previous-git-commit