

git

adam okulicz-kozaryn
adam.okulicz.kozaryn@gmail.com

this version: Friday 5th February, 2016 12:54

outline

misc

git

outline

misc

git

- ◇ as mentioned in my email...
- ◇ it is important to effectively do “diff”
(differentiate two versions of the same file)
- ◇ diff in npp

<http://www.addictivetips.com/windows-tips/>

[compare-two-source-codes-and-text-files-in-notepad-with-compare/](#)

- ◇ as mentioned in my email...
- ◇ it is important to effectively do “diff”
(differentiate two versions of the same file)

◇ diff in npp

<http://www.addictivetips.com/windows-tips/>

[compare-two-source-codes-and-text-files-in-notepad-with-compare/](#)

- ◇ as mentioned in my email...
- ◇ it is important to effectively do “diff”
(differentiate two versions of the same file)

◇ diff in npp

<http://www.addictivetips.com/windows-tips/>

[compare-two-source-codes-and-text-files-in-notepad-with-compare/](#)

- ◇ as mentioned in my email...
- ◇ it is important to effectively do “diff”
(differentiate two versions of the same file)
- ◇ diff in npp

<http://www.addictivetips.com/windows-tips/>

[compare-two-source-codes-and-text-files-in-notepad-with-compare/](#)

- ◇ as mentioned in my email...
- ◇ it is important to effectively do “diff”
(differentiate two versions of the same file)
- ◇ diff in npp

<http://www.addictivetips.com/windows-tips/>

[compare-two-source-codes-and-text-files-in-notepad-with-compare/](#)

- ◇ as mentioned in my email...
- ◇ it is important to effectively do “diff”
(differentiate two versions of the same file)
- ◇ diff in npp

`http://www.addictivetips.com/windows-tips/`

`compare-two-source-codes-and-text-files-in-notepad-with-compare/`

outline

misc

git

- ◇ again, git is made for tracking text files, not binaries (humans cannot read binaries)
 - e.g. see Sarah's and Rubias's repos with .gif and .dta
- ◇ fine if binaries are small (couple MB); otherwise load them from url
- ◇ or try git-submodule or git-annex
 - <http://stackoverflow.com/questions/540535/managing-large-binary-files-with-git>

- ◇ again, git is made for tracking text files, not binaries (humans cannot read binaries)
 - e.g. see Sarah's and Rubias's repos with .gif and .dta
- ◇ fine if binaries are small (couple MB); otherwise load them from url
- ◇ or try git-submodule or git-annex
 - <http://stackoverflow.com/questions/540535/managing-large-binary-files-with-git>

- ◇ again, git is made for tracking text files, not binaries (humans cannot read binaries)
 - e.g. see Sarah's and Rubias's repos with .gif and .dta
- ◇ fine if binaries are small (couple MB); otherwise load them from url
- ◇ or try git-submodule or git-annex

<http://stackoverflow.com/questions/540535/managing-large-binary-files-with-git>

- ◇ again, git is made for tracking text files, not binaries (humans cannot read binaries)
 - e.g. see Sarah's and Rubias's repos with .gif and .dta
- ◇ fine if binaries are small (couple MB); otherwise load them from url
- ◇ or try git-submodule or git-annex

<http://stackoverflow.com/questions/540535/managing-large-binary-files-with-git>

workflow in the class

- ◇ you also learn from each other
- ◇ there is user (student) generated content in git like in wikipedia or youtube

workflow in the class

- ◇ you also learn from each other
- ◇ there is user (student) generated content in git like in wikipedia or youtube

more git ref

◇ for mac users: <http://stackoverflow.com/questions/455698/best-visual-client-for-git-on-mac-os-x>

◇ a huge list of frontends

https://git.wiki.kernel.org/articles/i/n/t/Interfaces,_frontends,_and_tools.html

more git ref

- ◇ for mac users: <http://stackoverflow.com/questions/455698/best-visual-client-for-git-on-mac-os-x>
- ◇ a huge list of frontends
https://git.wiki.kernel.org/articles/i/n/t/Interfaces,_frontends,_and_tools.html

◇ again, there is a ton of git resources online, see links from previous classes

• one more link: installing, configuring, using, etc:
<http://book.git-scm.com/index.html>

• again, assignments are due in git, so at least you have to be able to push to class repo

◇ again, there is a ton of git resources online, see links from previous classes

- one more link: installing, configuring, using, etc:

<http://book.git-scm.com/index.html>

◇ again, assignments are due in git, so at least you have to be able to push to class repo

◇ again, there is a ton of git resources online, see links from previous classes

- one more link: installing, configuring, using, etc:

<http://book.git-scm.com/index.html>

◇ again, assignments are due in git, so at least you have to be able to push to class repo

- ◇ again, there is a ton of git resources online, see links from previous classes
- one more link: installing, configuring, using, etc:
<http://book.git-scm.com/index.html>
- ◇ again, assignments are due in git, so at least you have to be able to push to class repo

- ◇ several of you are already pushing to class repo, it's a great practice...
- ◇ it does take some experimenting
- ◇ don't be afraid to push and pull things around
- ◇ e.g. you can copy your classmate's repo... e.g.:
 - `git clone http://aok.us.to/cgi-bin/cgit.cgi/julie/`
- ◇ note that you can also push a folder with files:
 - `http://aok.us.to/cgi-bin/cgit.cgi/melissa/`

- ◇ several of you are already pushing to class repo, it's a great practice...
- ◇ it does take some experimenting
- ◇ don't be afraid to push and pull things around
- ◇ e.g. you can copy your classmate's repo... e.g.:
 - `git clone http://aok.us.to/cgi-bin/cgit.cgi/julie/`
- ◇ note that you can also push a folder with files:
 - `http://aok.us.to/cgi-bin/cgit.cgi/melissa/`

- ◇ several of you are already pushing to class repo, it's a great practice...
- ◇ it does take some experimenting
- ◇ don't be afraid to push and pull things around
- ◇ e.g. you can copy your classmate's repo... e.g.:
 - `git clone http://aok.us.to/cgi-bin/cgit.cgi/julie/`
- ◇ note that you can also push a folder with files:
 - `http://aok.us.to/cgi-bin/cgit.cgi/melissa/`

- ◇ several of you are already pushing to class repo, it's a great practice...
- ◇ it does take some experimenting
- ◇ don't be afraid to push and pull things around
- ◇ e.g. you can copy your classmate's repo... e.g.:
 - `git clone http://aok.us.to/cgi-bin/cgit.cgi/julie/`
- ◇ note that you can also push a folder with files:
 - `http://aok.us.to/cgi-bin/cgit.cgi/melissa/`

- ◇ several of you are already pushing to class repo, it's a great practice...
- ◇ it does take some experimenting
- ◇ don't be afraid to push and pull things around
- ◇ e.g. you can copy your classmate's repo... e.g.:
 - `git clone http://aok.us.to/cgi-bin/cgit.cgi/julie/`
- ◇ note that you can also push a folder with files:
 - `http://aok.us.to/cgi-bin/cgit.cgi/melissa/`

- ◇ several of you are already pushing to class repo, it's a great practice...
- ◇ it does take some experimenting
- ◇ don't be afraid to push and pull things around
- ◇ e.g. you can copy your classmate's repo... e.g.:
 - `git clone http://aok.us.to/cgi-bin/cgit.cgi/julie/`
- ◇ note that you can also push a folder with files:
 - `http://aok.us.to/cgi-bin/cgit.cgi/melissa/`

- ◇ several of you are already pushing to class repo, it's a great practice...
- ◇ it does take some experimenting
- ◇ don't be afraid to push and pull things around
- ◇ e.g. you can copy your classmate's repo... e.g.:
 - `git clone http://aok.us.to/cgi-bin/cgit.cgi/julie/`
- ◇ note that you can also push a folder with files:
 - `http://aok.us.to/cgi-bin/cgit.cgi/melissa/`

git collaboration

- ◇ you can also try to collaborate on the code with your classmates...
- ◇ this will be a handy skill if you work on assignments in groups
- ◇ still, you can just have commits by one person and credit for everybody
 - (e.g. people sitting at one machine and brainstorming the code)
- ◇ of course, do not try to push anything to somebody's else repo unless that person gives you permission

git collaboration

- ◇ you can also try to collaborate on the code with your classmates...
- ◇ this will be a handy skill if you work on assignments in groups
- ◇ still, you can just have commits by one person and credit for everybody
 - (e.g. people sitting at one machine and brainstorming the code)
- ◇ of course, do not try to push anything to somebody's else repo unless that person gives you permission

git collaboration

- ◇ you can also try to collaborate on the code with your classmates...
- ◇ this will be a handy skill if you work on assignments in groups
- ◇ still, you can just have commits by one person and credit for everybody
 - (e.g. people sitting at one machine and brainstorming the code)
- ◇ of course, do not try to push anything to somebody's else repo unless that person gives you permission

git collaboration

- ◇ you can also try to collaborate on the code with your classmates...
- ◇ this will be a handy skill if you work on assignments in groups
- ◇ still, you can just have commits by one person and credit for everybody
 - (e.g. people sitting at one machine and brainstorming the code)
- ◇ of course, do not try to push anything to somebody's else repo unless that person gives you permission

git collaboration

- ◇ you can also try to collaborate on the code with your classmates...
- ◇ this will be a handy skill if you work on assignments in groups
- ◇ still, you can just have commits by one person and credit for everybody
 - (e.g. people sitting at one machine and brainstorming the code)
- ◇ of course, do not try to push anything to somebody's else repo unless that person gives you permission

outline

misc

git

git

- ◇ you did great with cross-commenting in git
- ◇ but class git repo is a mess
- ◇ if i start adding my comments on your files and on comments it will be even a bigger mess
- ◇ please do not push anything until after tomorrow
- ◇ i will figure out a way to clean it up
 - possibly slicing commits into branches
- ◇ so that it is easier to read
- ◇ any ideas from you? let's see few readable commits in Paula's repo, and more commits and also 2 branches in Julie's repo
- ◇ for now, given our current git workflow, few tips...

git

- ◇ you did great with cross-commenting in git
- ◇ but class git repo is a mess
- ◇ if i start adding my comments on your files and on comments it will be even a bigger mess
- ◇ please do not push anything until after tomorrow
- ◇ i will figure out a way to clean it up
 - possibly slicing commits into branches
- ◇ so that it is easier to read
- ◇ any ideas from you? let's see few readable commits in Paula's repo, and more commits and also 2 branches in Julie's repo
- ◇ for now, given our current git workflow, few tips...

git

- ◇ you did great with cross-commenting in git
- ◇ but class git repo is a mess
- ◇ if i start adding my comments on your files and on comments it will be even a bigger mess
- ◇ please do not push anything until after tomorrow
- ◇ i will figure out a way to clean it up
 - possibly slicing commits into branches
- ◇ so that it is easier to read
- ◇ any ideas from you? let's see few readable commits in Paula's repo, and more commits and also 2 branches in Julie's repo
- ◇ for now, given our current git workflow, few tips...

git

- ◇ you did great with cross-commenting in git
- ◇ but class git repo is a mess
- ◇ if i start adding my comments on your files and on comments it will be even a bigger mess
- ◇ please do not push anything until after tomorrow
- ◇ i will figure out a way to clean it up
 - possibly slicing commits into branches
- ◇ so that it is easier to read
- ◇ any ideas from you? let's see few readable commits in Paula's repo, and more commits and also 2 branches in Julie's repo
- ◇ for now, given our current git workflow, few tips...

git

- ◇ you did great with cross-commenting in git
- ◇ but class git repo is a mess
- ◇ if i start adding my comments on your files and on comments it will be even a bigger mess
- ◇ please do not push anything until after tomorrow
- ◇ i will figure out a way to clean it up
 - possibly slicing commits into branches
- ◇ so that it is easier to read
- ◇ any ideas from you? let's see few readable commits in Paula's repo, and more commits and also 2 branches in Julie's repo
- ◇ for now, given our current git workflow, few tips...

git

- ◇ you did great with cross-commenting in git
- ◇ but class git repo is a mess
- ◇ if i start adding my comments on your files and on comments it will be even a bigger mess
- ◇ please do not push anything until after tomorrow
- ◇ i will figure out a way to clean it up
 - possibly slicing commits into branches
- ◇ so that it is easier to read
- ◇ any ideas from you? let's see few readable commits in Paula's repo, and more commits and also 2 branches in Julie's repo
- ◇ for now, given our current git workflow, few tips...

git

- ◇ you did great with cross-commenting in git
- ◇ but class git repo is a mess
- ◇ if i start adding my comments on your files and on comments it will be even a bigger mess
- ◇ please do not push anything until after tomorrow
- ◇ i will figure out a way to clean it up
 - possibly slicing commits into branches
- ◇ so that it is easier to read
- ◇ any ideas from you? let's see few readable commits in Paula's repo, and more commits and also 2 branches in Julie's repo
- ◇ for now, given our current git workflow, few tips...

git

- ◇ you did great with cross-commenting in git
- ◇ but class git repo is a mess
- ◇ if i start adding my comments on your files and on comments it will be even a bigger mess
- ◇ please do not push anything until after tomorrow
- ◇ i will figure out a way to clean it up
 - possibly slicing commits into branches
- ◇ so that it is easier to read
- ◇ any ideas from you? let's see few readable commits in Paula's repo, and more commits and also 2 branches in Julie's repo
- ◇ for now, given our current git workflow, few tips...

git

- ◇ you did great with cross-commenting in git
- ◇ but class git repo is a mess
- ◇ if i start adding my comments on your files and on comments it will be even a bigger mess
- ◇ please do not push anything until after tomorrow
- ◇ i will figure out a way to clean it up
 - possibly slicing commits into branches
- ◇ so that it is easier to read
- ◇ any ideas from you? let's see few readable commits in Paula's repo, and more commits and also 2 branches in Julie's repo
- ◇ for now, given our current git workflow, few tips...

- ◇ with cross-commenting we bumped into merging problem
- it happens when there are two versions of the same file that are conflicting
- say, A edited B's file and pushed it
- at the same time B edited the same (B's) file and pulled A's edits
- that creates a conflict that you need to resolve
- git will tell you that "automatic merge failed"

- ◇ with cross-commenting we bumped into merging problem
- it happens when there are two versions of the same file that are conflicting
- say, A edited B's file and pushed it
- at the same time B edited the same (B's) file and pulled A's edits
- that creates a conflict that you need to resolve
- git will tell you that "automatic merge failed"

- ◇ with cross-commenting we bumped into merging problem
- it happens when there are two versions of the same file that are conflicting
- say, A edited B's file and pushed it
- at the same time B edited the same (B's) file and pulled A's edits
- that creates a conflict that you need to resolve
- git will tell you that "automatic merge failed"

- ◇ with cross-commenting we bumped into merging problem
 - it happens when there are two versions of the same file that are conflicting
 - say, A edited B's file and pushed it
 - at the same time B edited the same (B's) file and pulled A's edits
 - that creates a conflict that you need to resolve
 - git will tell you that "automatic merge failed"

- ◇ with cross-commenting we bumped into merging problem
- it happens when there are two versions of the same file that are conflicting
- say, A edited B's file and pushed it
- at the same time B edited the same (B's) file and pulled A's edits
- that creates a conflict that you need to resolve
- git will tell you that "automatic merge failed"

- ◇ with cross-commenting we bumped into merging problem
 - it happens when there are two versions of the same file that are conflicting
 - say, A edited B's file and pushed it
 - at the same time B edited the same (B's) file and pulled A's edits
 - that creates a conflict that you need to resolve
 - git will tell you that "automatic merge failed"

◇ git would put conflicting two versions of a file into one file

· and conflicts will be delimited by

```
<<<<<<< HEAD : README.txt
```

```
meeplib for CSE 491, 2012, at MSU. YYY something else
```

```
=====
```

```
meeplib for CSE 491, 2012, at MSU. XXXXX conflict!!!
```

```
>>>>>>> conflict : README.txt
```

◇ merge two files by manually editing the combined file

· and by removing git delimiters

· and commit the resulting file

◇ and there are some specialized tools to do the merge;

· see SO for discussion

· <http://stackoverflow.com/questions/137102/whats-the-best-visual-merge-tool-for-git>

git

- ◆ git would put conflicting two versions of a file into one file
- and conflicts will be delimited by

<<<<<<< *HEAD : README.txt*

meeplib for CSE 491, 2012, at MSU. YYY something else!

=====

meeplib for CSE 491, 2012, at MSU. XXXXX conflict!!!

>>>>>>> *conflict : README.txt*

- merge two files by manually editing the combined file
- and by removing git delimiters
- and commit the resulting file
- and there are some specialized tools to do the merge;
- see SO for discussion

<http://stackoverflow.com/questions/137102/whats-the-best-visual-merge-tool-for-git>

git

- ◆ git would put conflicting two versions of a file into one file
- and conflicts will be delimited by

<<<<<<< *HEAD : README.txt*

meeplib for CSE 491, 2012, at MSU. YYY something else!

=====

meeplib for CSE 491, 2012, at MSU. XXXXX conflict!!!

>>>>>>> *conflict : README.txt*

- merge two files by manually editing the combined file
- and by removing git delimiters
- and commit the resulting file
- and there are some specialized tools to do the merge;
- see SO for discussion

<http://stackoverflow.com/questions/137102/whats-the-best-visual-merge-tool-for-git>

git

- git would put conflicting two versions of a file into one file
- and conflicts will be delimited by

```
<<<<<<< HEAD : README.txt
```

```
meeplib for CSE 491, 2012, at MSU. YYY something else!
```

```
=====
```

```
meeplib for CSE 491, 2012, at MSU. XXXXX conflict!!!
```

```
>>>>>>> conflict : README.txt
```

- merge two files by manually editing the combined file
- and by removing git delimiters
- and commit the resulting file
- and there are some specialized tools to do the merge;
- see SO for discussion

<http://stackoverflow.com/questions/137102/whats-the-best-visual-merge-tool-for-git>

git

- ◇ git would put conflicting two versions of a file into one file
 - and conflicts will be delimited by

```
<<<<<<< HEAD : README.txt
meeplib for CSE 491, 2012, at MSU. YYY something else!
=====
meeplib for CSE 491, 2012, at MSU. XXXXX conflict!!!
>>>>>>> conflict : README.txt
```
- ◇ merge two files by manually editing the combined file
- ◇ and by removing git delimiters
- ◇ and commit the resulting file
- ◇ and there are some specialized tools to do the merge;
- ◇ see SO for discussion

<http://stackoverflow.com/questions/137102/whats-the-best-visual-merge-tool-for-git>

git

- ◇ git would put conflicting two versions of a file into one file
 - and conflicts will be delimited by

```
<<<<<<< HEAD : README.txt
meeplib for CSE 491, 2012, at MSU. YYY something else!
=====
meeplib for CSE 491, 2012, at MSU. XXXXX conflict!!!
>>>>>>> conflict : README.txt
```
- ◇ merge two files by manually editing the combined file
- ◇ and by removing git delimiters
- ◇ and commit the resulting file
- ◇ and there are some specialized tools to do the merge;
- ◇ see SO for discussion

<http://stackoverflow.com/questions/137102/whats-the-best-visual-merge-tool-for-git>

git

- ◇ git would put conflicting two versions of a file into one file
 - and conflicts will be delimited by

```
<<<<<<< HEAD : README.txt
meeplib for CSE 491, 2012, at MSU. YYY something else!
=====
meeplib for CSE 491, 2012, at MSU. XXXXX conflict!!!
>>>>>>> conflict : README.txt
```
- ◇ merge two files by manually editing the combined file
- ◇ and by removing git delimiters
- ◇ and commit the resulting file
- ◇ and there are some specialized tools to do the merge;
- ◇ see SO for discussion

<http://stackoverflow.com/questions/137102/whats-the-best-visual-merge-tool-for-git>

git

- ◇ git would put conflicting two versions of a file into one file
 - and conflicts will be delimited by

```
<<<<<<< HEAD : README.txt
meeplib for CSE 491, 2012, at MSU. YYY something else!
=====
meeplib for CSE 491, 2012, at MSU. XXXXX conflict!!!
>>>>>>> conflict : README.txt
```
- ◇ merge two files by manually editing the combined file
 - and by removing git delimiters
 - and commit the resulting file
- ◇ and there are some specialized tools to do the merge;
 - see SO for discussion

* <http://stackoverflow.com/questions/137102/whats-the-best-visual-merge-tool-for-git>

git

- ◇ git would put conflicting two versions of a file into one file
 - and conflicts will be delimited by

```
<<<<<<< HEAD : README.txt
meeplib for CSE 491, 2012, at MSU. YYY something else!
=====
meeplib for CSE 491, 2012, at MSU. XXXXX conflict!!!
>>>>>>> conflict : README.txt
```
- ◇ merge two files by manually editing the combined file
 - and by removing git delimiters
 - and commit the resulting file
- ◇ and there are some specialized tools to do the merge;
 - see SO for discussion

* <http://stackoverflow.com/questions/137102/whats-the-best-visual-merge-tool-for-git>

git

- ◇ git would put conflicting two versions of a file into one file
 - and conflicts will be delimited by

```
<<<<<<< HEAD : README.txt
meeplib for CSE 491, 2012, at MSU. YYY something else!
=====
meeplib for CSE 491, 2012, at MSU. XXXXX conflict!!!
>>>>>>> conflict : README.txt
```
- ◇ merge two files by manually editing the combined file
 - and by removing git delimiters
 - and commit the resulting file
- ◇ and there are some specialized tools to do the merge;
 - see SO for discussion

*<http://stackoverflow.com/questions/137102/whats-the-best-visual-merge-tool-for-git>

git

- ◇ git would put conflicting two versions of a file into one file
 - and conflicts will be delimited by

```
<<<<<<< HEAD : README.txt
meeplib for CSE 491, 2012, at MSU. YYY something else!
=====
meeplib for CSE 491, 2012, at MSU. XXXXX conflict!!!
>>>>>>> conflict : README.txt
```
- ◇ merge two files by manually editing the combined file
 - and by removing git delimiters
 - and commit the resulting file
- ◇ and there are some specialized tools to do the merge;
 - see SO for discussion

* <http://stackoverflow.com/questions/137102/whats-the-best-visual-merge-tool-for-git>

git

- ◇ git would put conflicting two versions of a file into one file
 - and conflicts will be delimited by

```
<<<<<<< HEAD : README.txt
meeplib for CSE 491, 2012, at MSU. YYY something else!
=====
meeplib for CSE 491, 2012, at MSU. XXXXX conflict!!!
>>>>>>> conflict : README.txt
```
- ◇ merge two files by manually editing the combined file
 - and by removing git delimiters
 - and commit the resulting file
- ◇ and there are some specialized tools to do the merge;
 - see SO for discussion

* <http://stackoverflow.com/questions/137102/whats-the-best-visual-merge-tool-for-git>

git

- ◇ git would put conflicting two versions of a file into one file
 - and conflicts will be delimited by

```
<<<<<<< HEAD : README.txt
meeplib for CSE 491, 2012, at MSU. YYY something else!
=====
meeplib for CSE 491, 2012, at MSU. XXXXX conflict!!!
>>>>>>> conflict : README.txt
```
- ◇ merge two files by manually editing the combined file
 - and by removing git delimiters
 - and commit the resulting file
- ◇ and there are some specialized tools to do the merge;
 - see SO for discussion

• <http://stackoverflow.com/questions/137102/whats-the-best-visual-merge-tool-for-git>

git under the hood

◇ <http://eagain.net/articles/git-for-computer-scientists/>

◇ <http://marklodato.github.com/visual-git-guide/index-en.html>

git under the hood

- ◇ <http://eagain.net/articles/git-for-computer-scientists/>
- ◇ <http://marklodato.github.com/visual-git-guide/index-en.html>