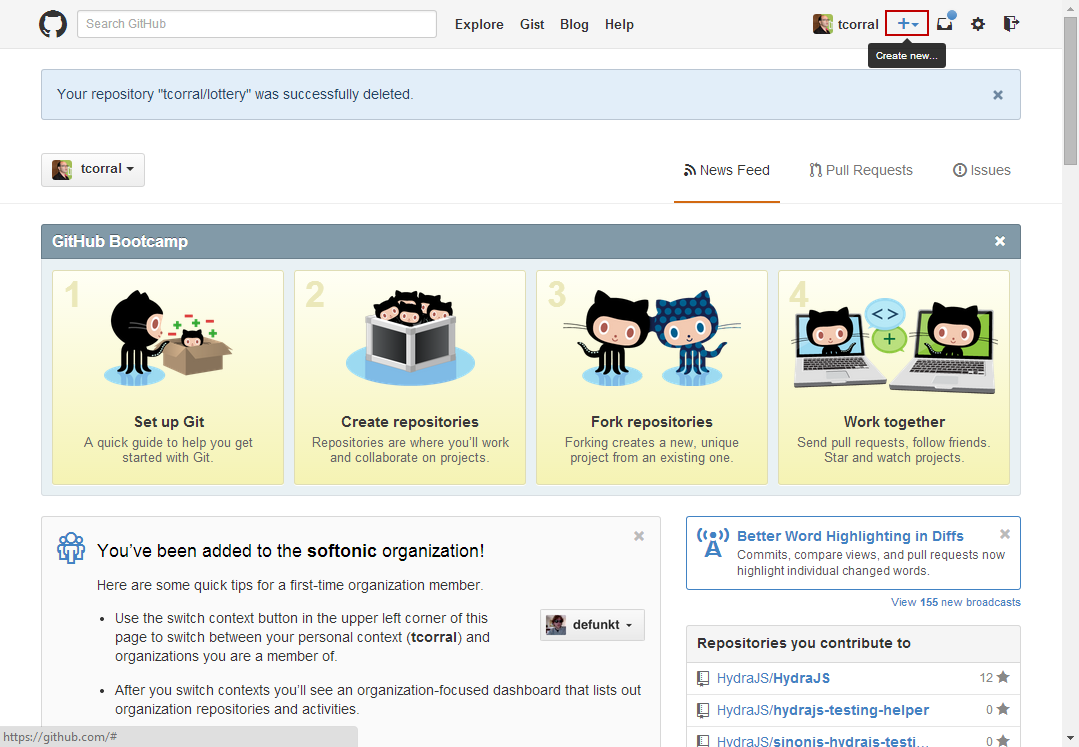
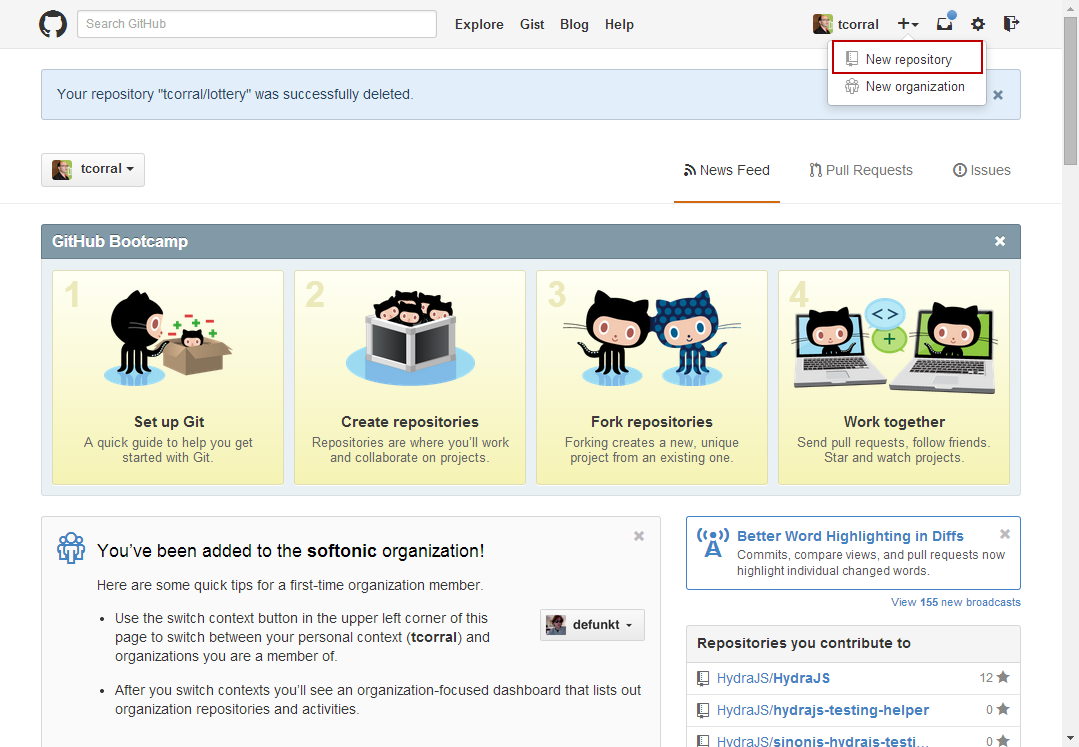
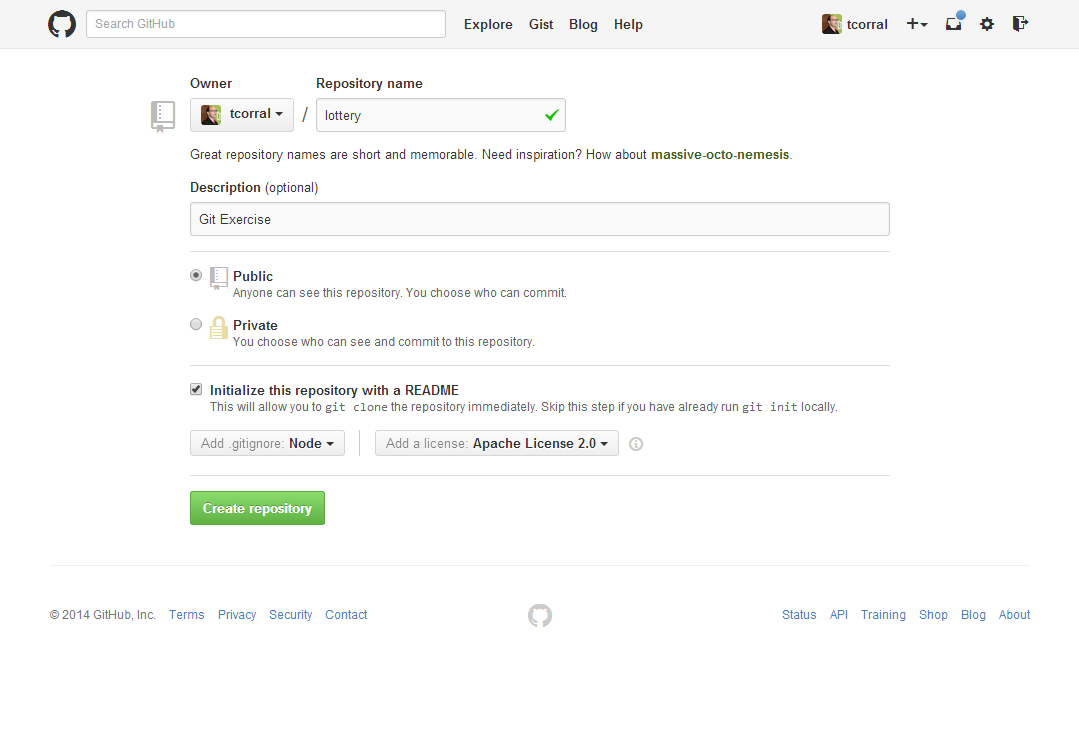
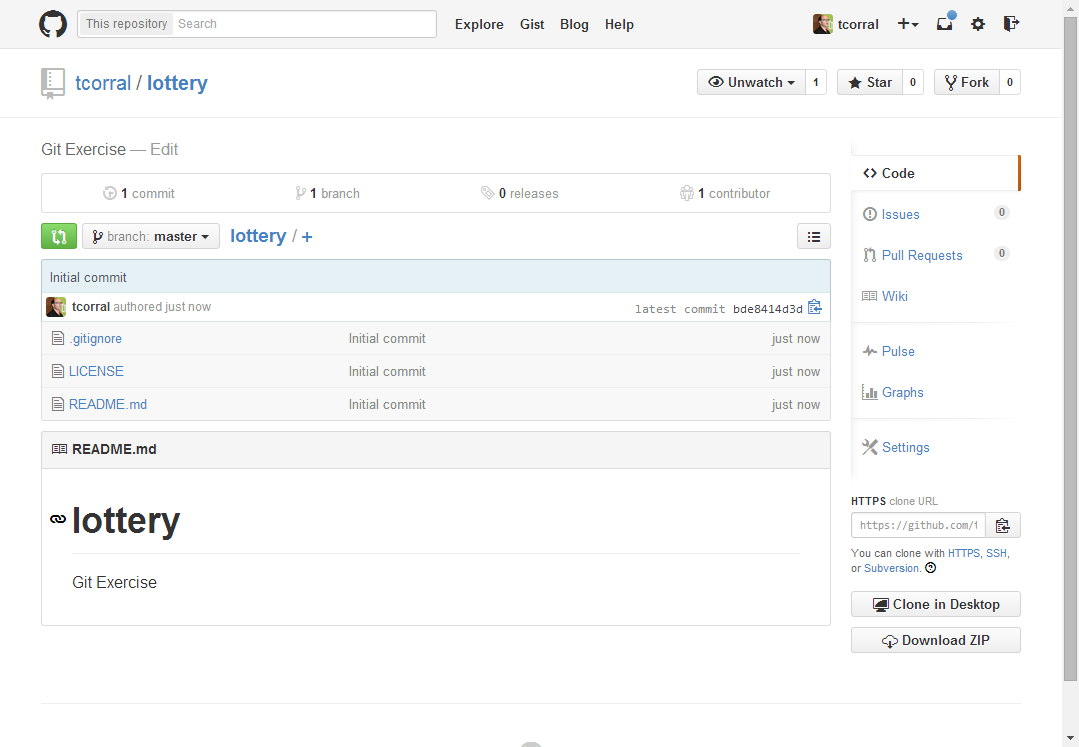
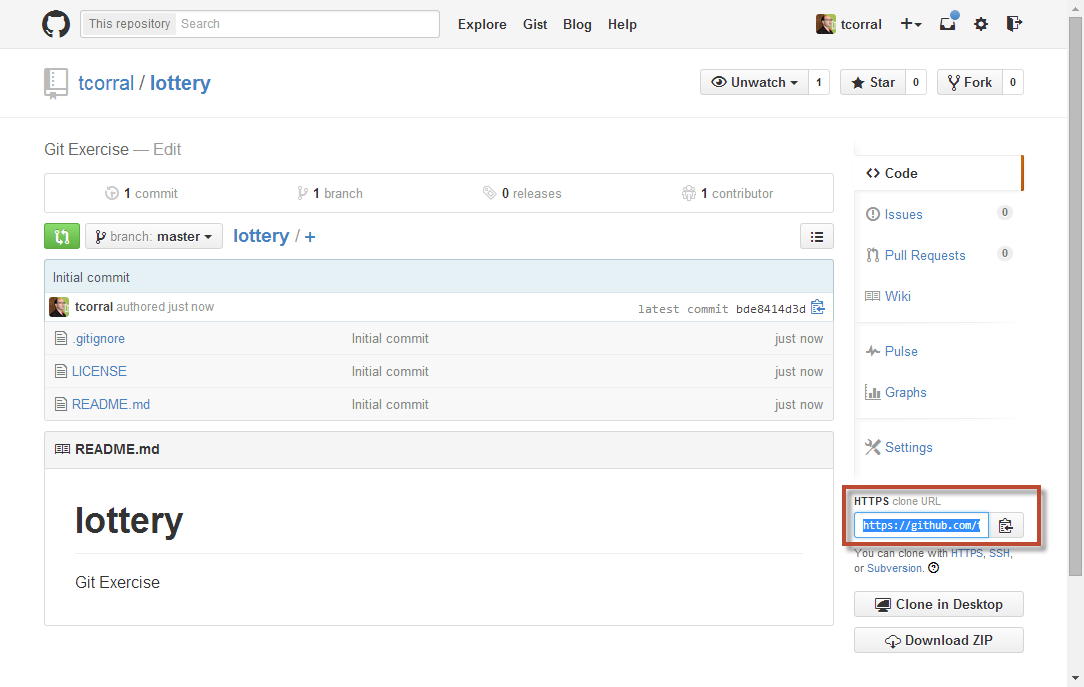
Create a repository in Github:



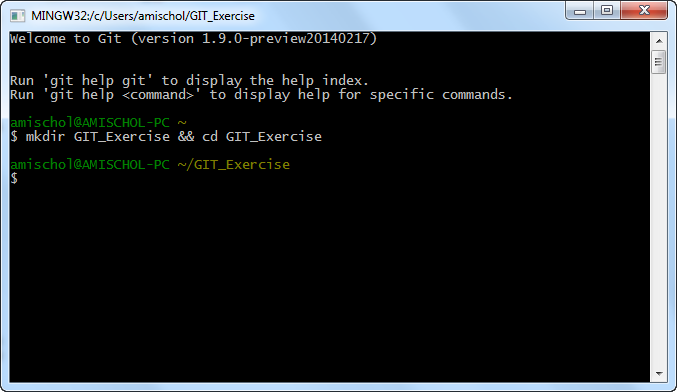


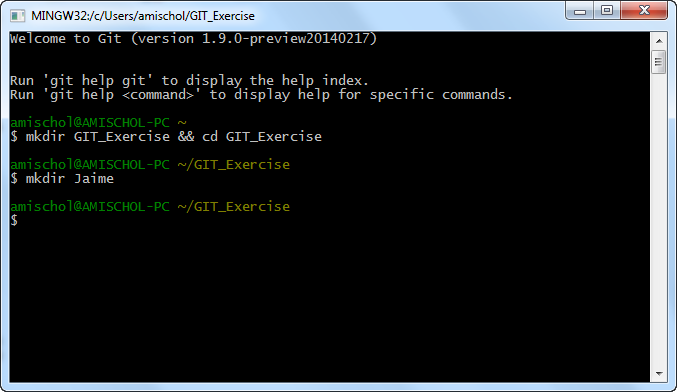


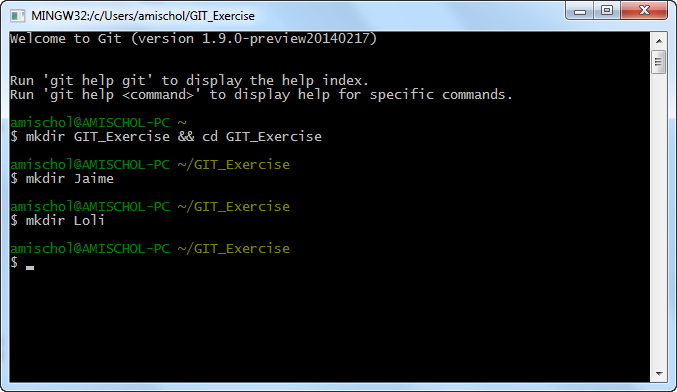


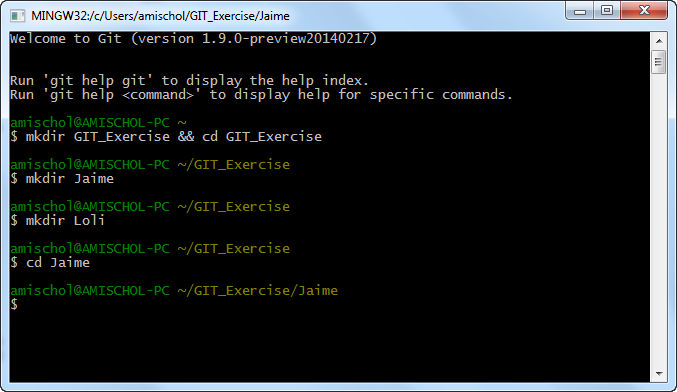


Create a folder to store the two users interaction simulation:

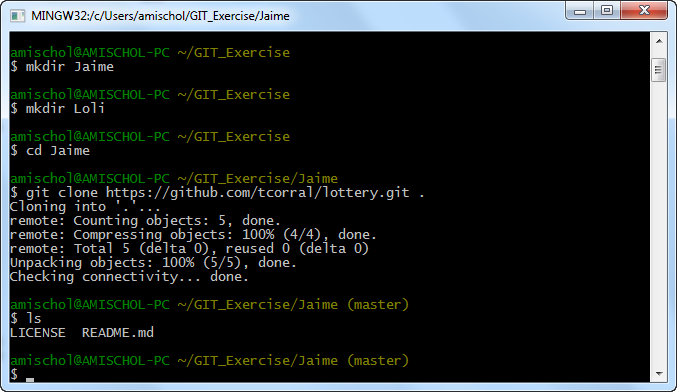


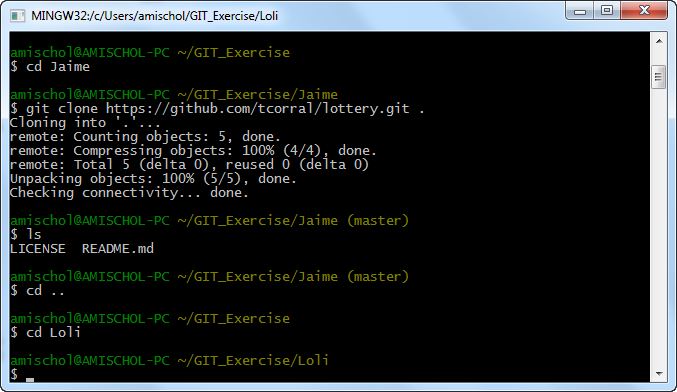


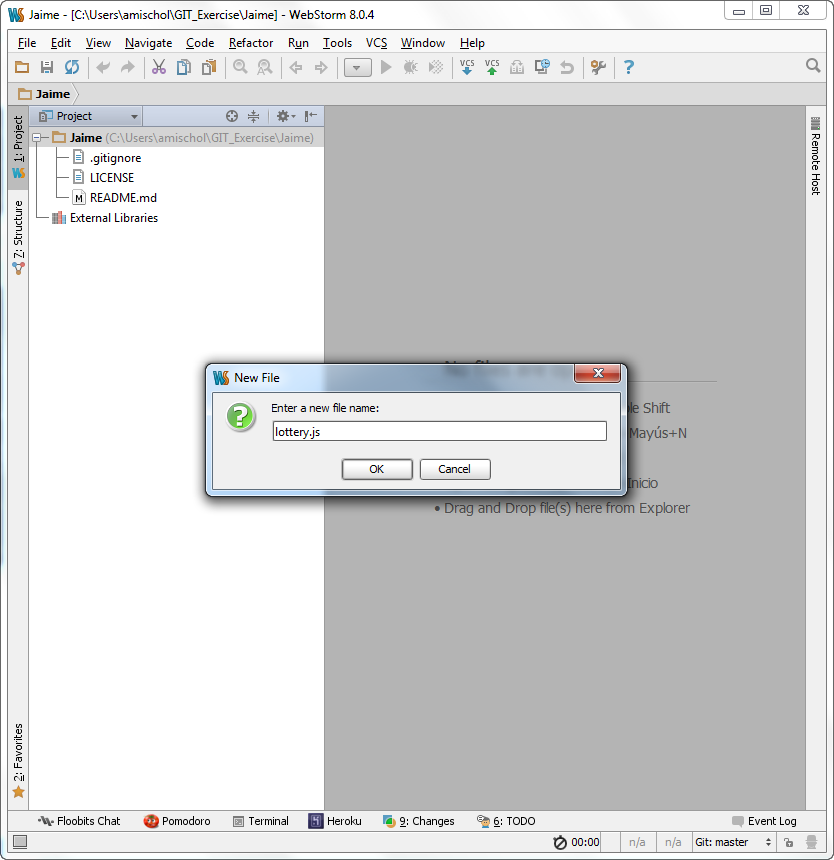


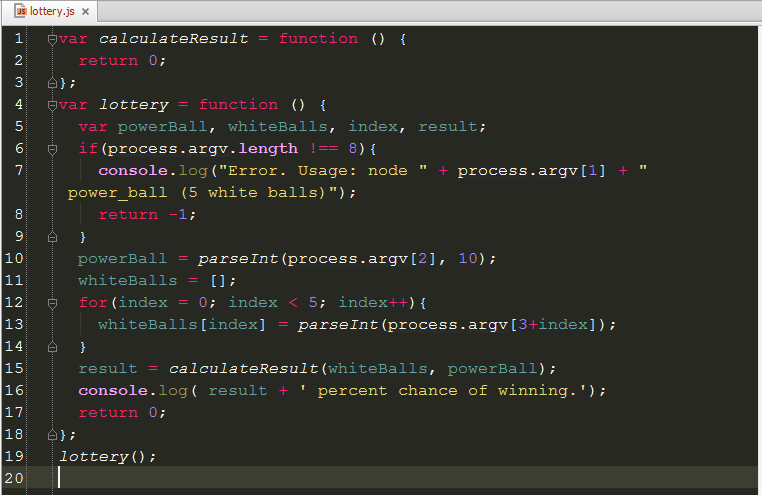


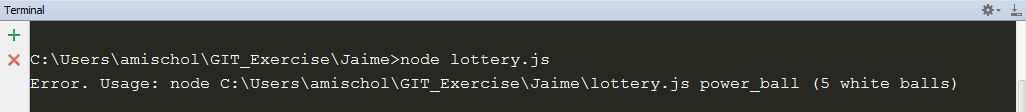
Clone the repo in the first user folder:

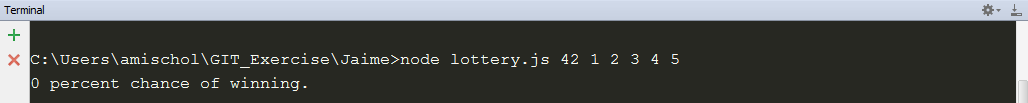


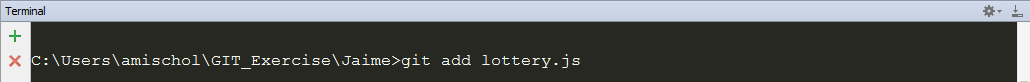


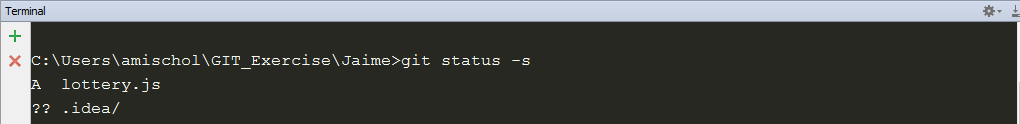


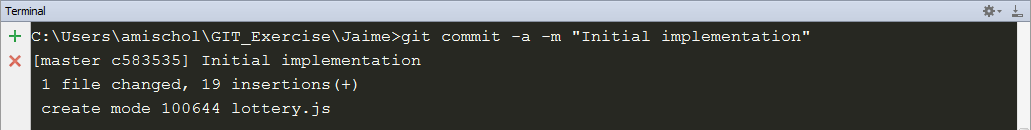




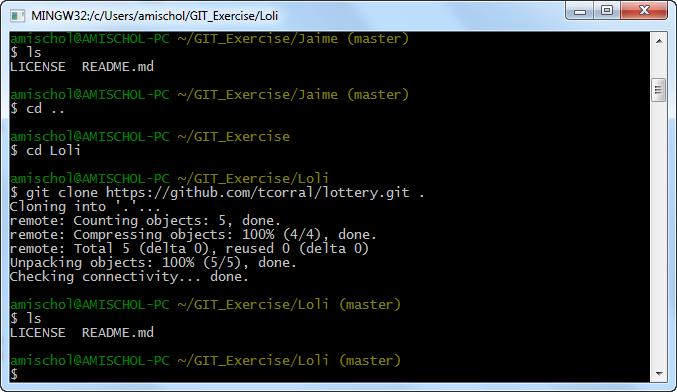








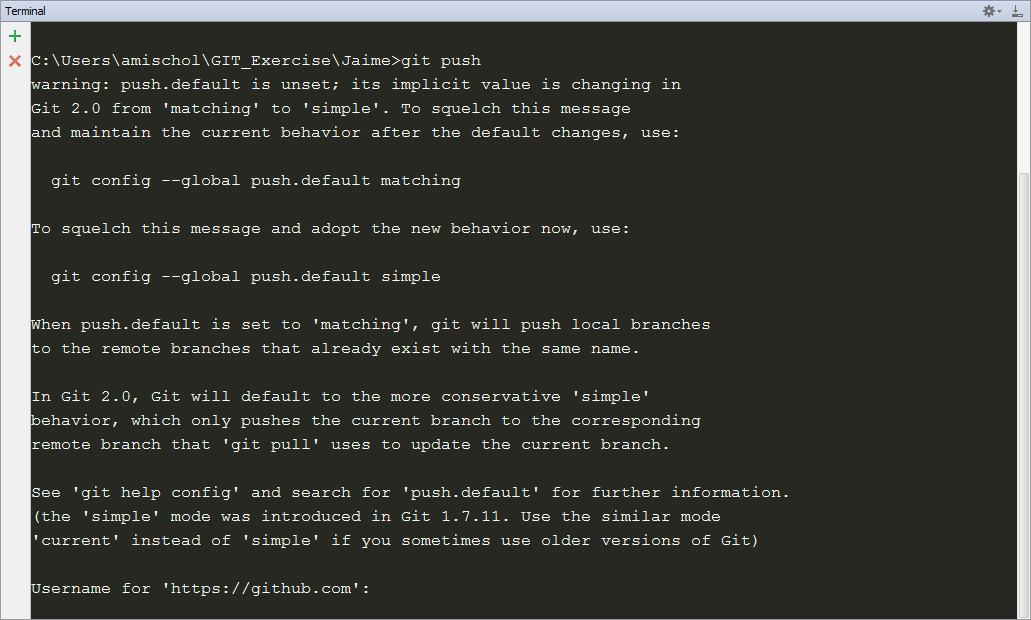
Clone the repo by the second user:

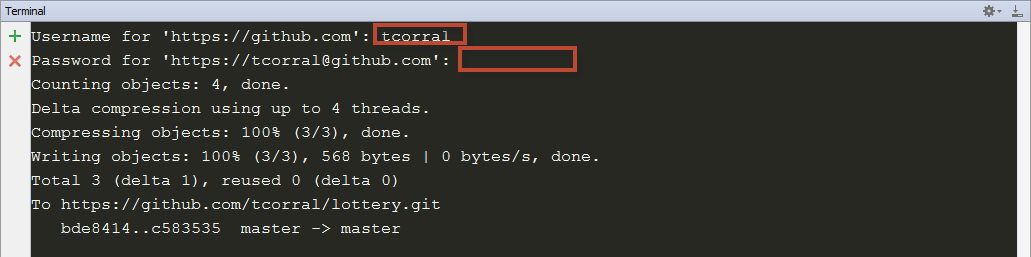


As you can see something weird happened, the lottery.js file doesn’t exist in the remote repository. Why?

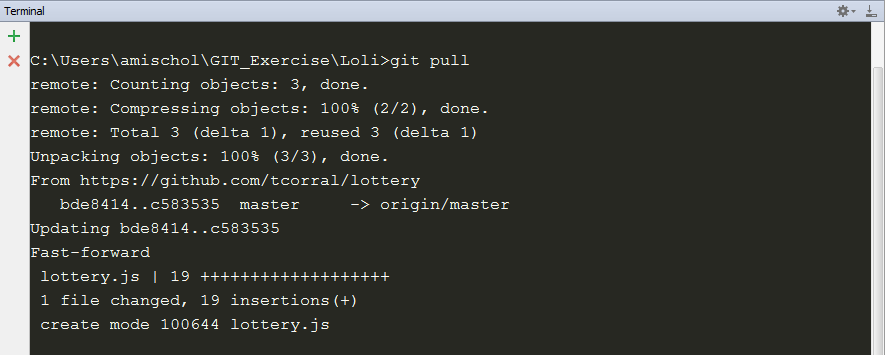
The problem is that in Git you need to commit and push in two different steps and Jaime forgot to push the changes to the remote git server.

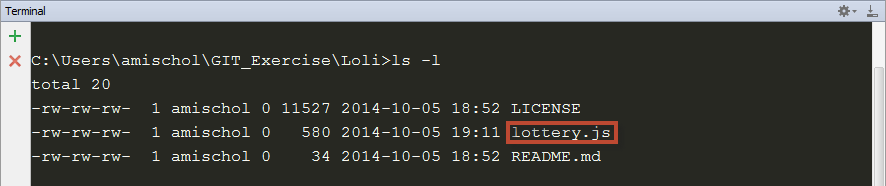
So we tell Jaime to push his changes.



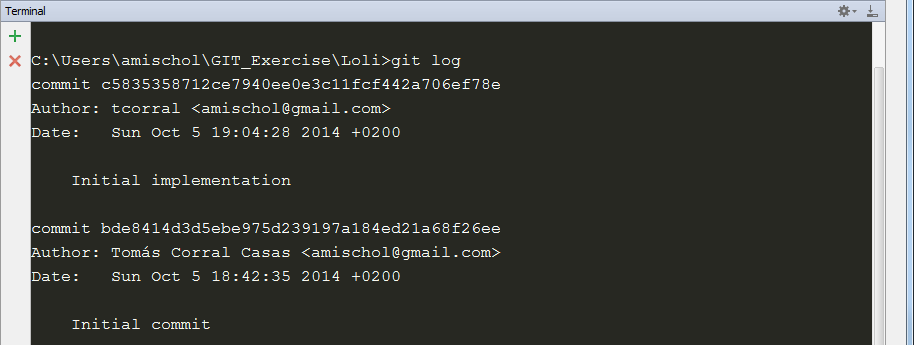


Now Loli could make pull to get the changes from Jaime.

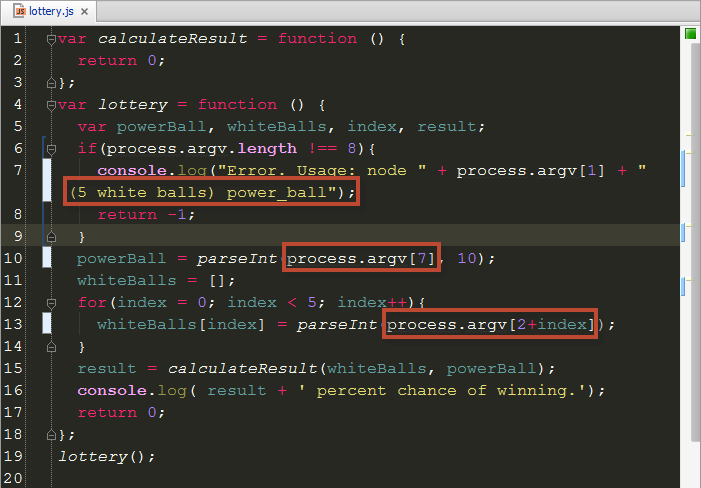


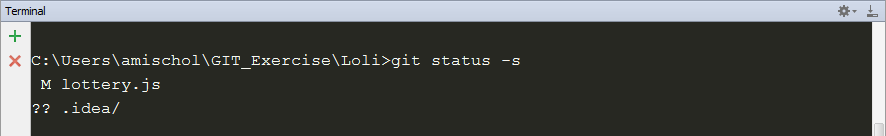


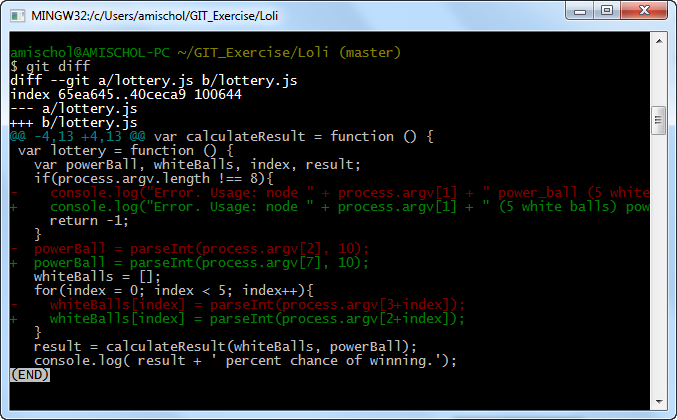
After getting the list of files Loli wants to know the log of changes:



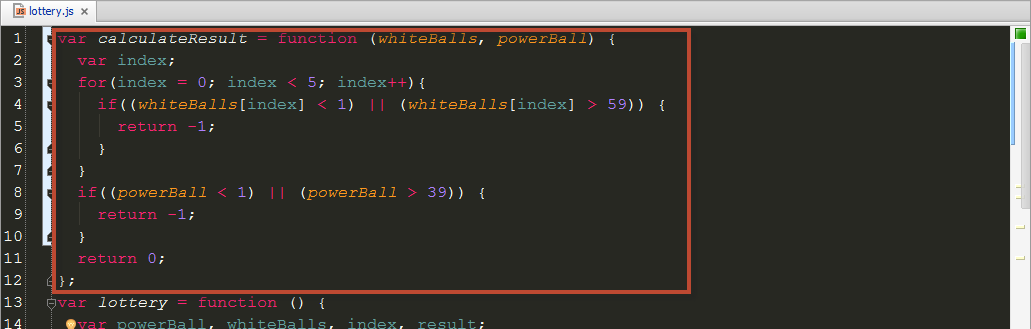
After checking the code Loli detected that the arguments are passed in a different order that should be fixed. The power ball should be the last argument and the white balls the first argument, so Loli goes to fix this code.

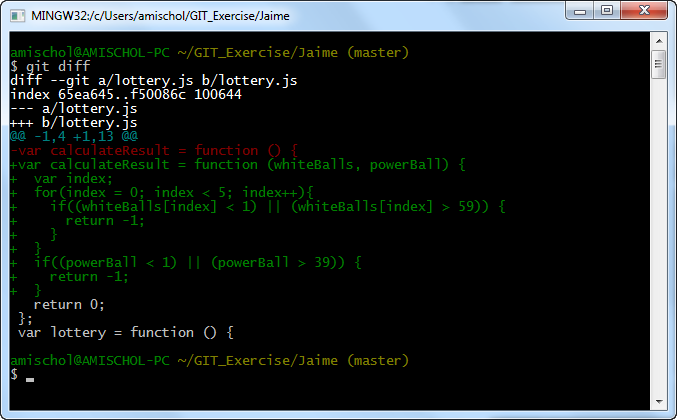




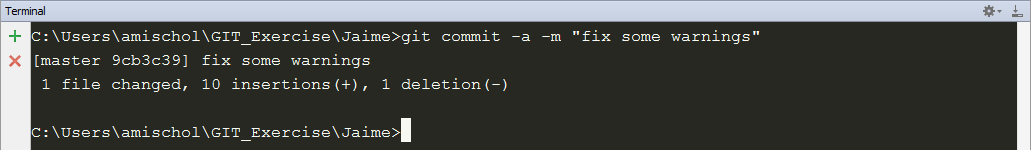


Meanwhile, Jaime has been coding as well, and it’s fixing a problem with the function calculateResult, it’s important to check arguments for validity. The power ball can be from 1 to 39 inclusive. The white balls can be 1 to 59 inclusive. So that he implements the error checking.

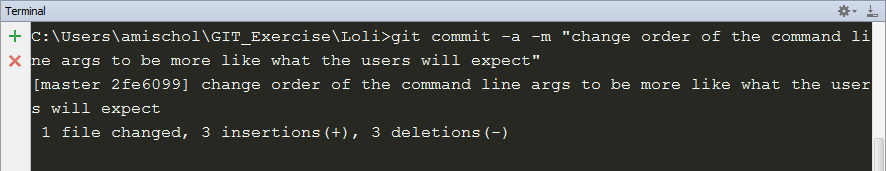


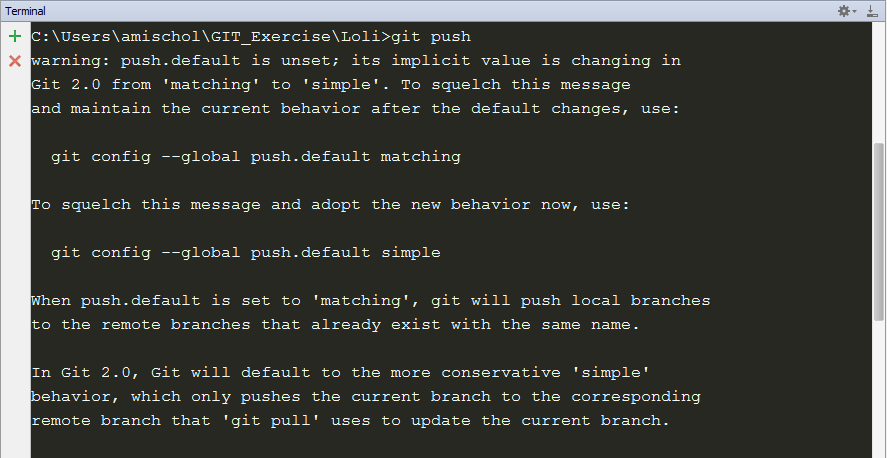
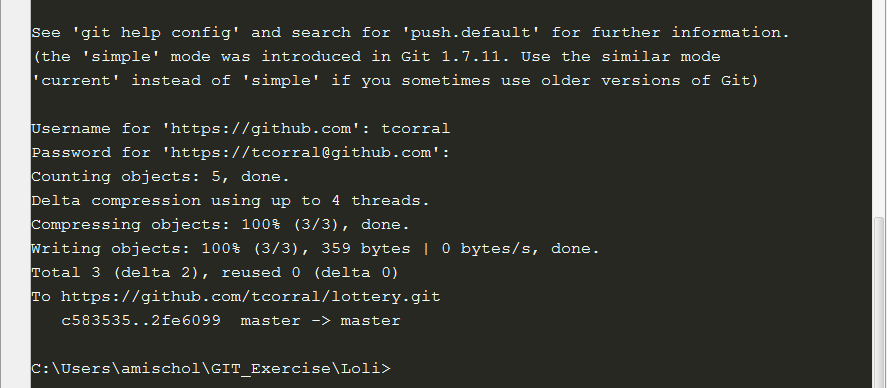


Now he makes the commit

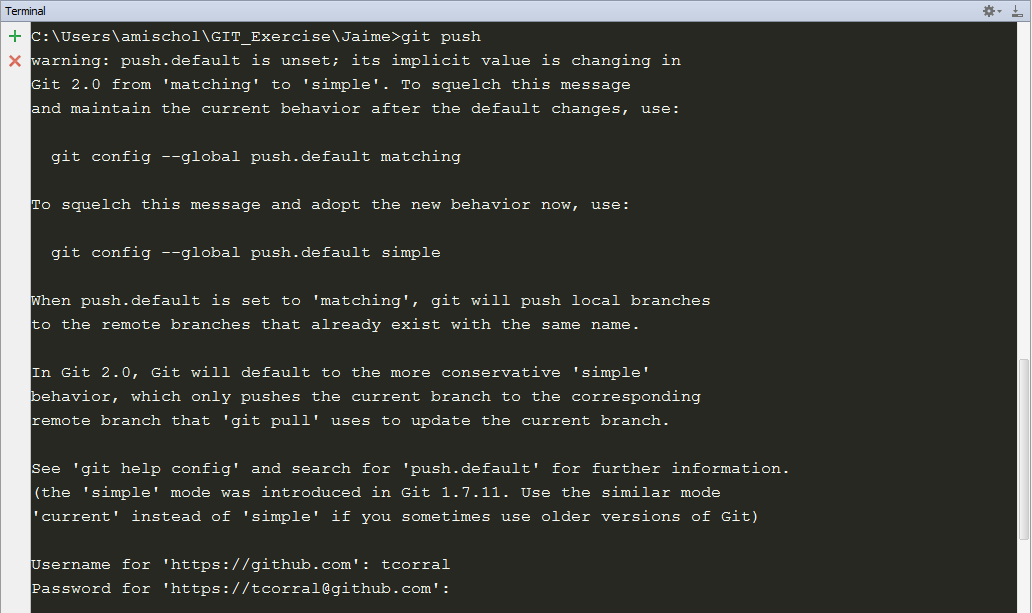


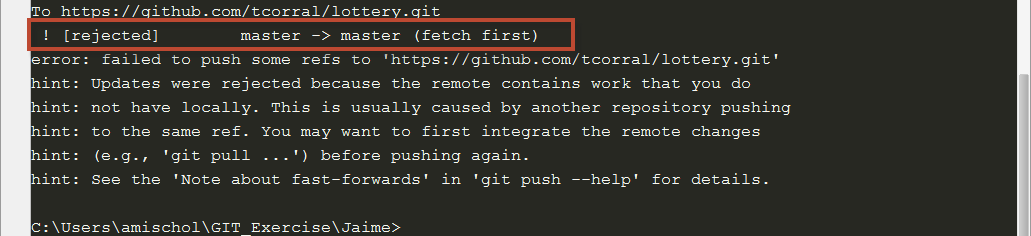
But when he is going to make the push Loli has commited and pushed her changes:



So when Jaime wants to push his changes something doesn’t works as expected:

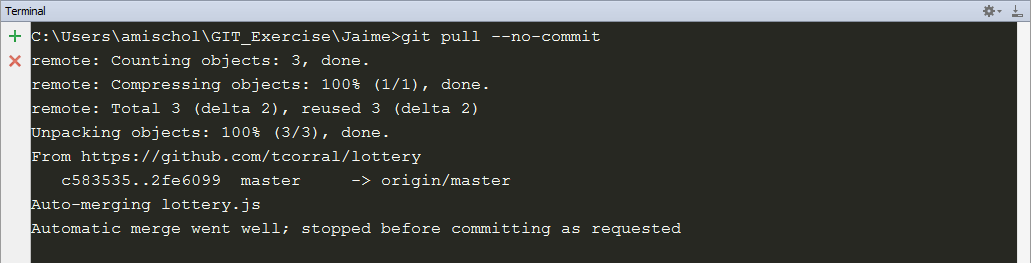


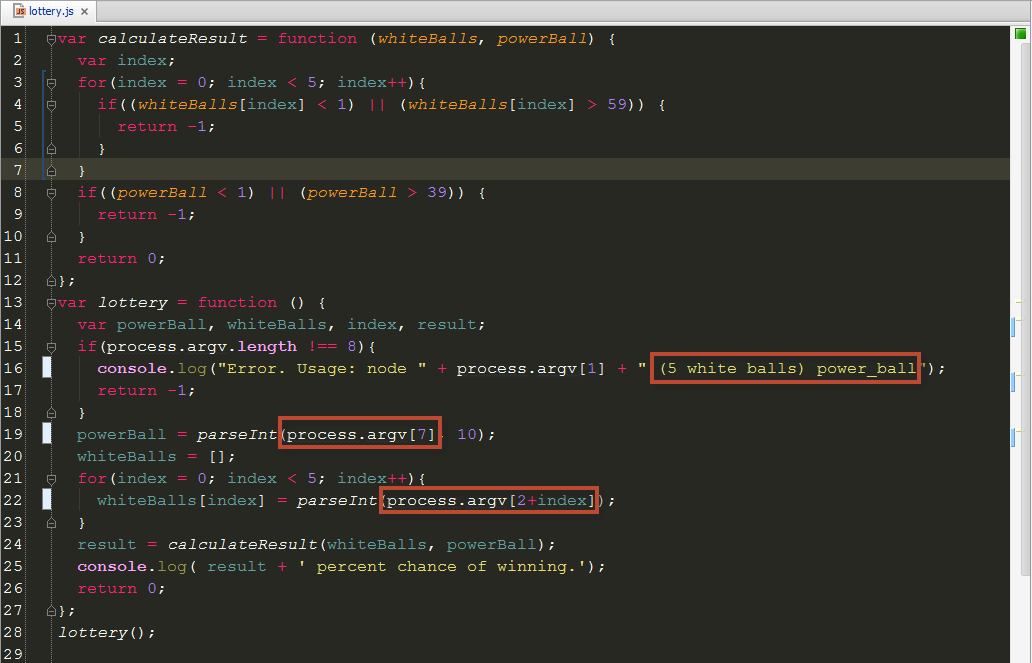


The push has been rejected due that there are remote changes that are different from local changes.

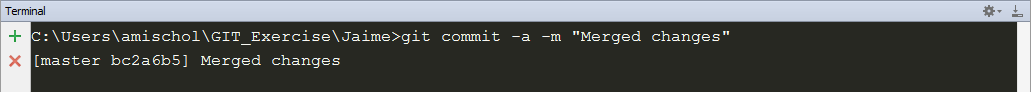
Jaime needs to download the changes from the remote repository so we are going how he will do it.

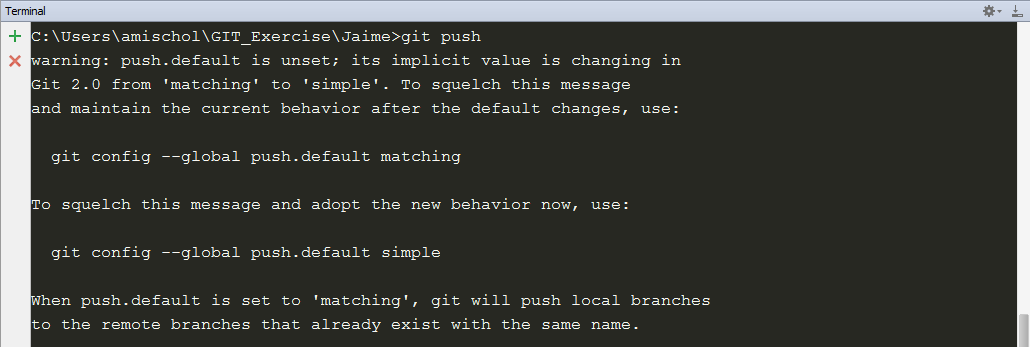
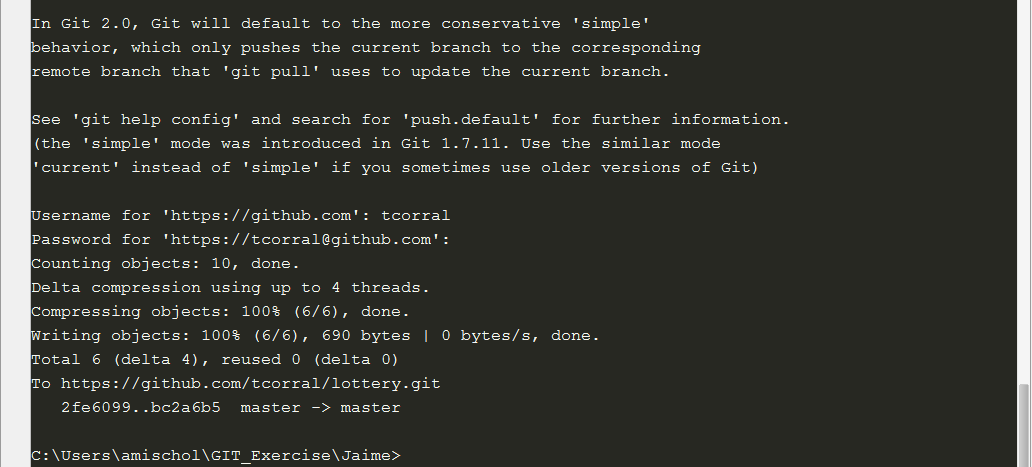
The command we should use is git pull but we have to note that git pull merges the changes (if it can) and commits the change, it could not be cool do it without check if the changes are what we expect, so we are going to use git pull –no-commit let us see how.

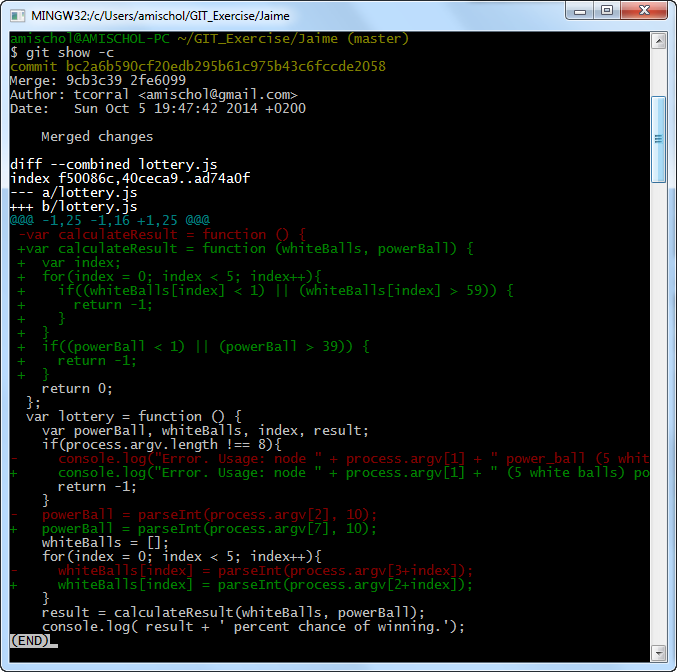




Jaime check the changes and sees that are ok, so Jaime commits these changes.



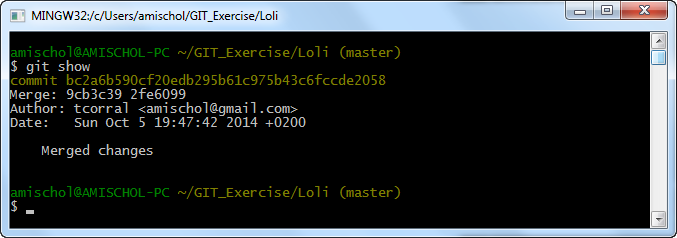
 

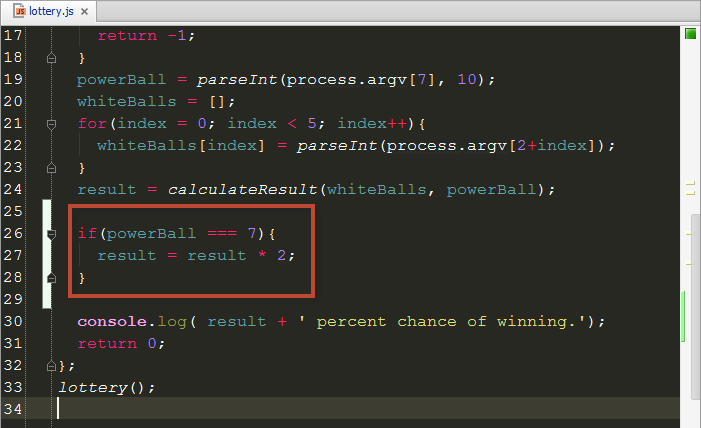


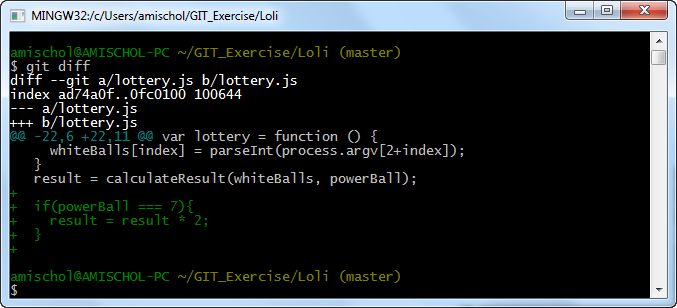
Meanwhile Loli is adding a new feature that was requested by the sales team: If the user chooses the lucky number 7 as the red ball (powerball), the chances of winning are doubled.

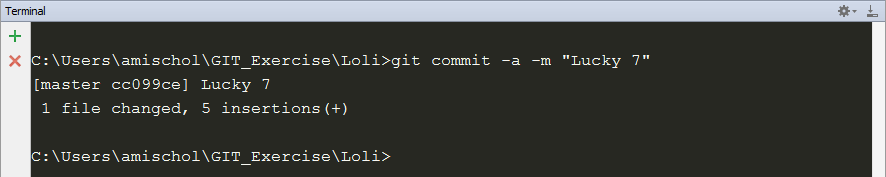
So that she is starting a new task she decides to begin with pull and update to make sure se has the latest code.

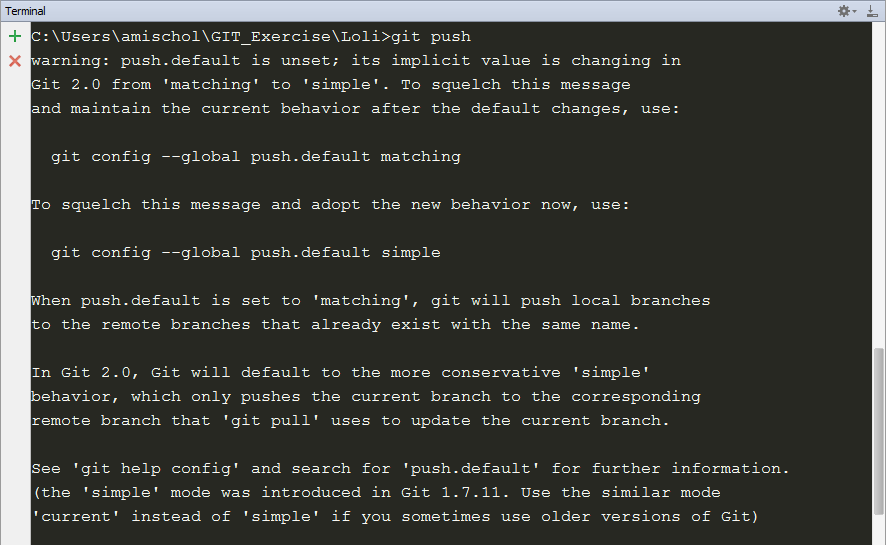
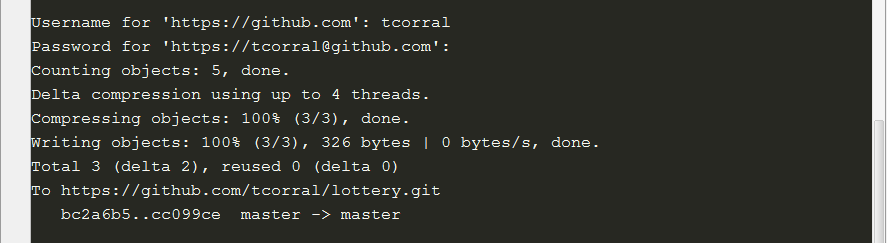




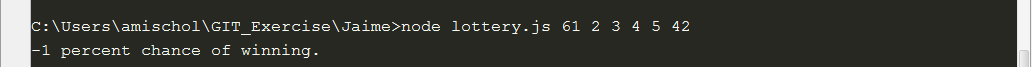




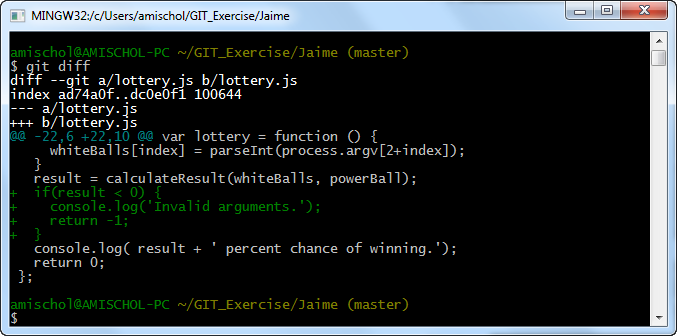


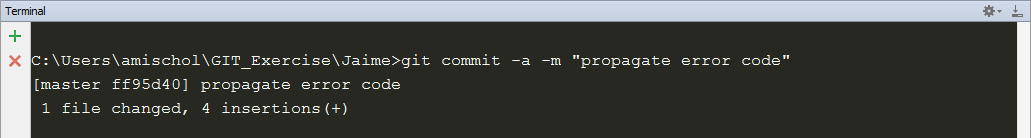
Meanwhile Jaime has realized his last hange had a bug. He modified calculateResult to return -1 for invalid arguments but he forgot to modify the caller to handle the error. As a consequence entering a ball number that is out of range causes the program to behave improperly.

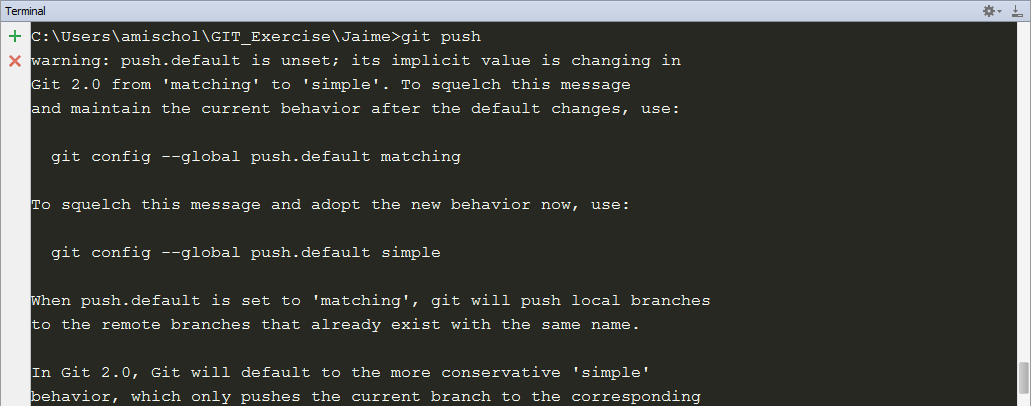
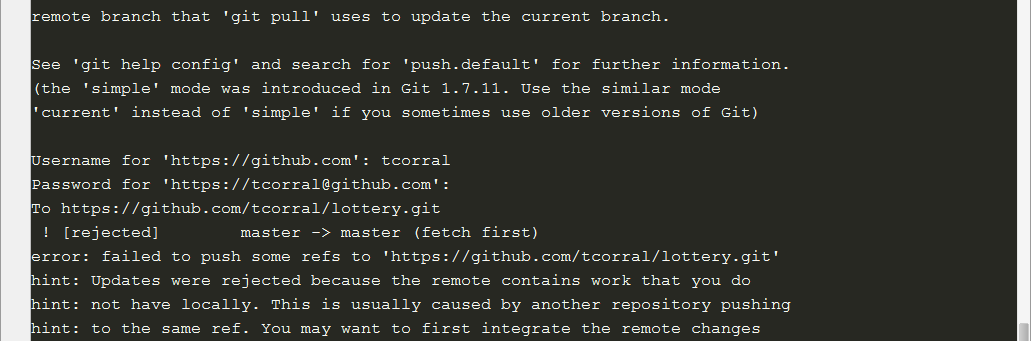
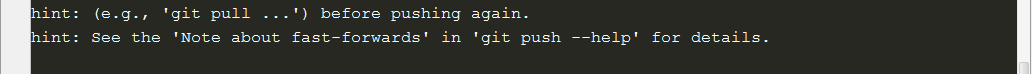


Ther percentage of winning can’t be a negative number. So Jaime adds an extra check for this case.

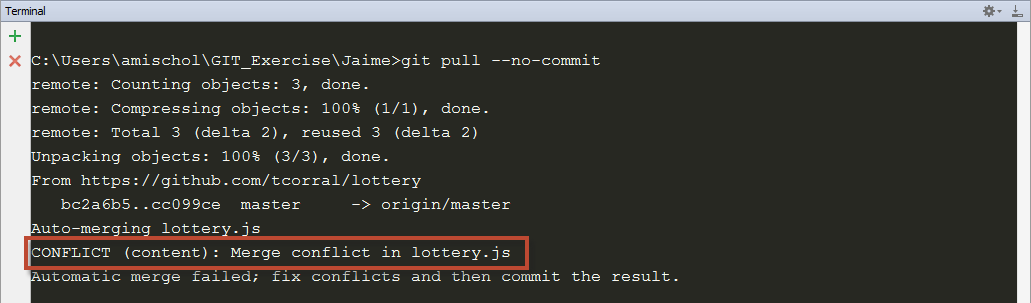


So proceeds to commit and push the fix.



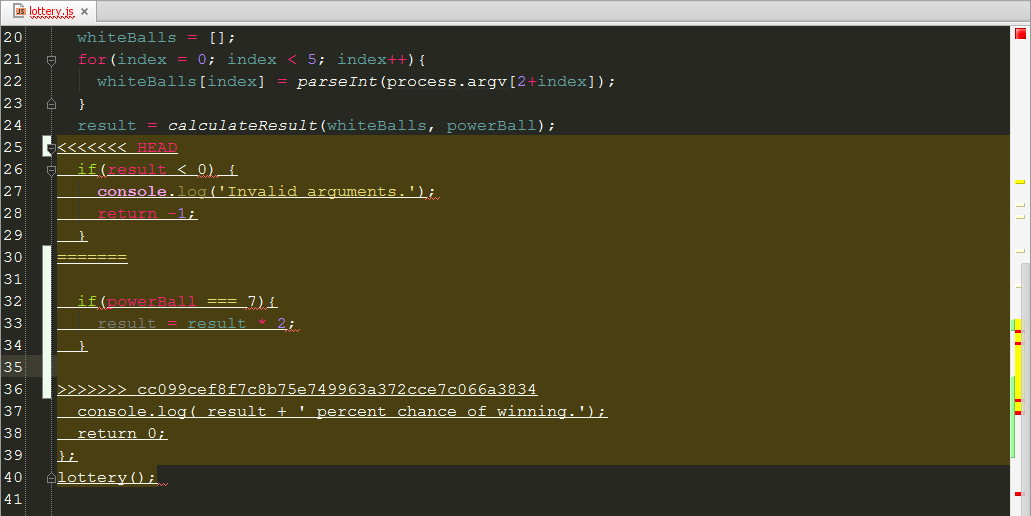
  

Oh no!! Again we have some problem when pushing the code could be that Loli have pushed a new changeset already. Jaime needs to pull and merge the changes with his own.



Umm! This is a big issue because it can’t be merged automatically it has some conflicts that should be resolved.

Check the code:

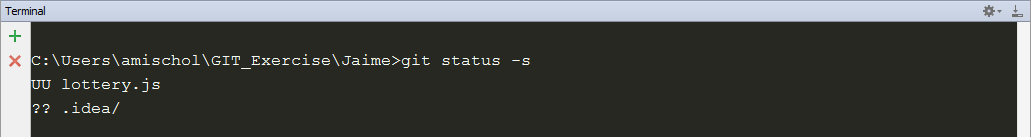


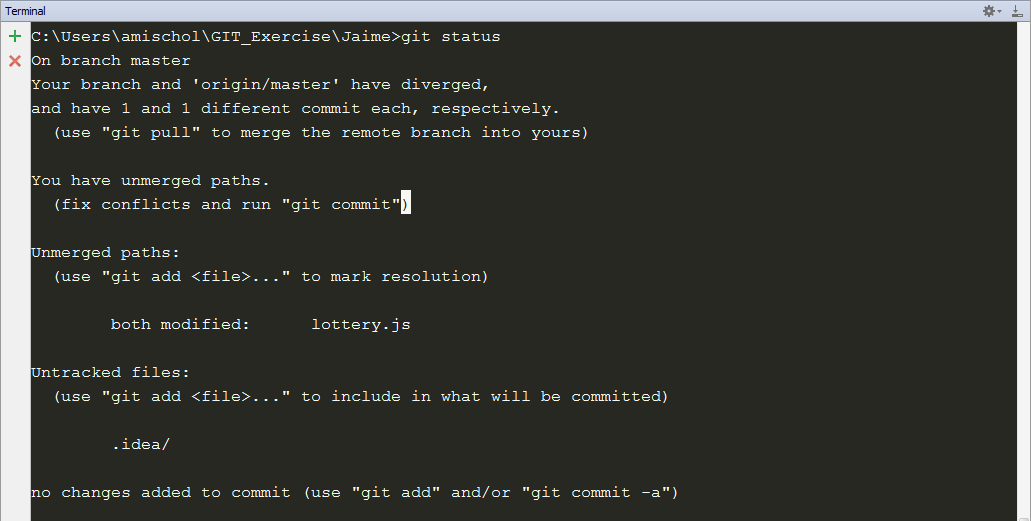
Git has included both Jaime and Loli’s code with conflict markers to delimit things. To fix this conflict we could add Loli’s code just after Jaime’s error checking code.

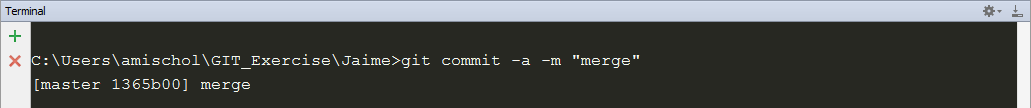


Check that it works as expected:

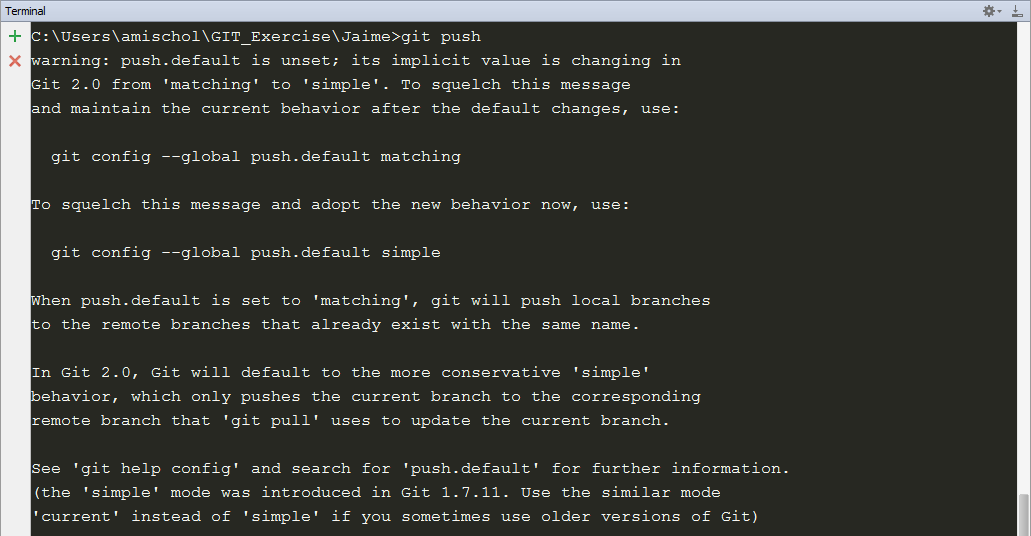
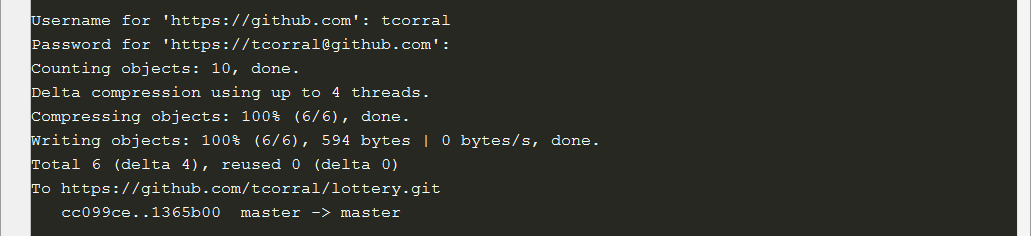




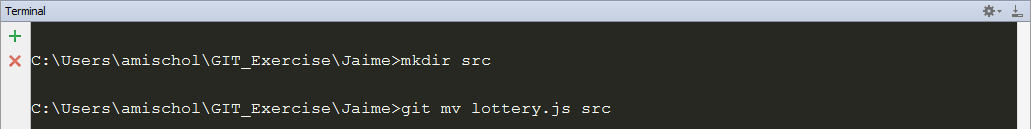




Good now Jaume tries again the push:

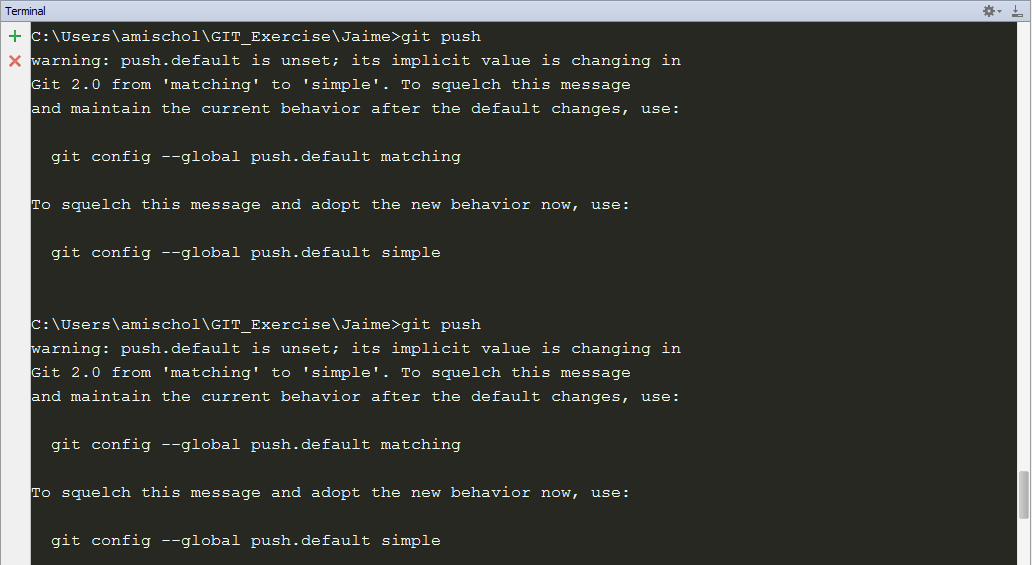
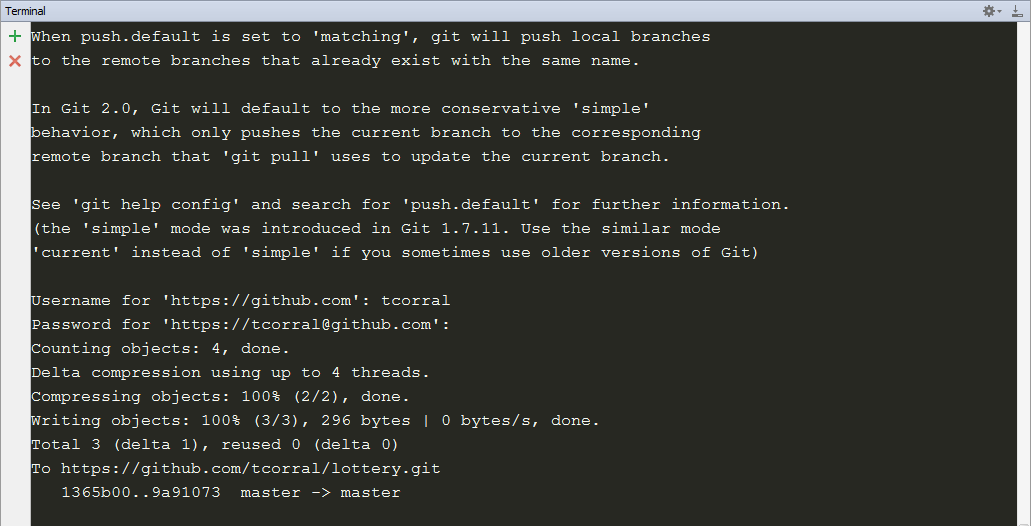
 

Harry wants to re-estructure the tree, to avoid to get too clutered the top level of the repository he decides to move their vast number of source code files in a src subdirectory.

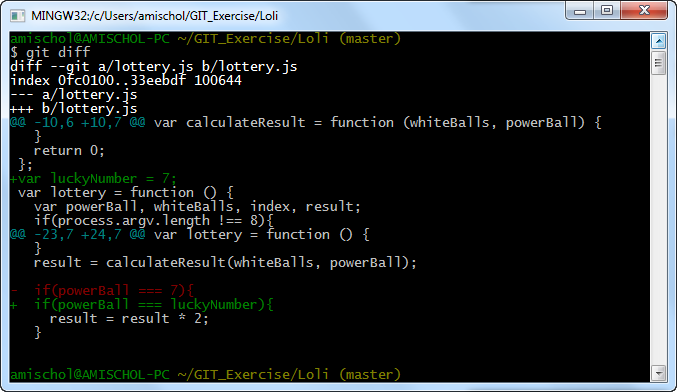




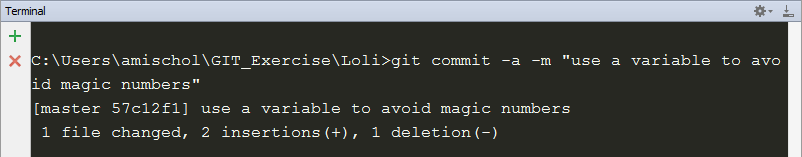


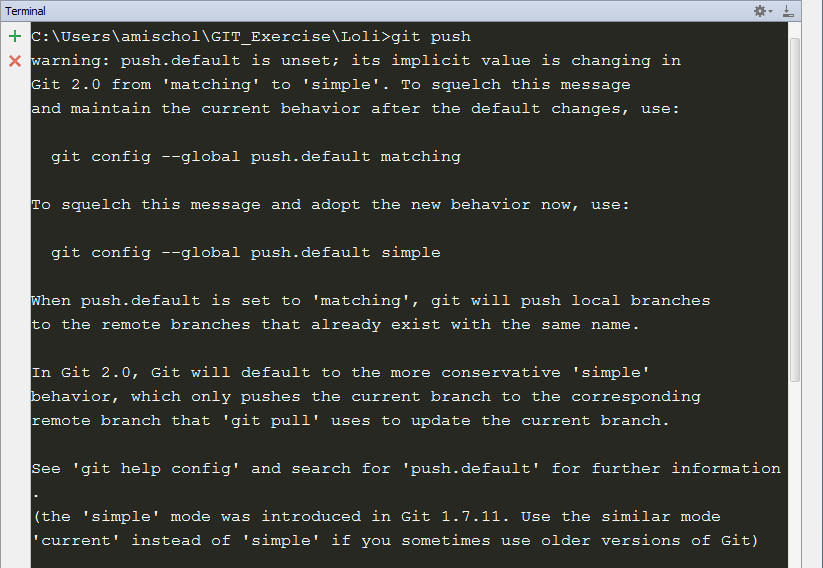
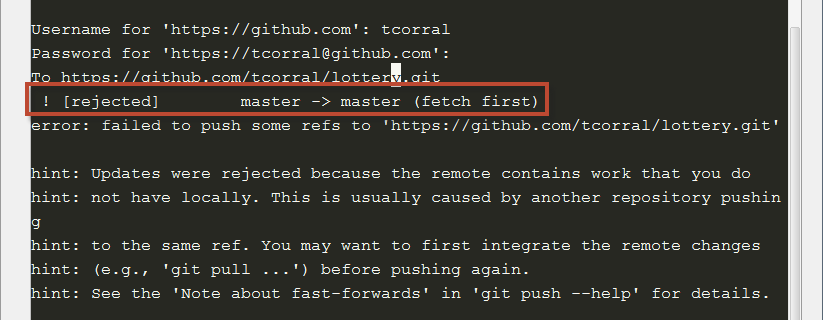
 

Loli is reviewing the code and she doesn’t likes to have the number 7 as a magic number so she decides to set it as a variable with a more clear name.

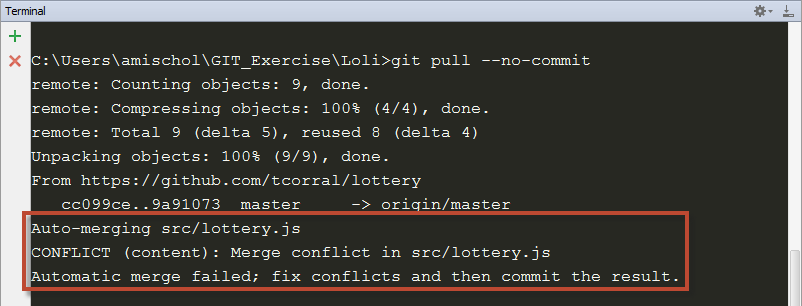


And immediately commits and push the change.

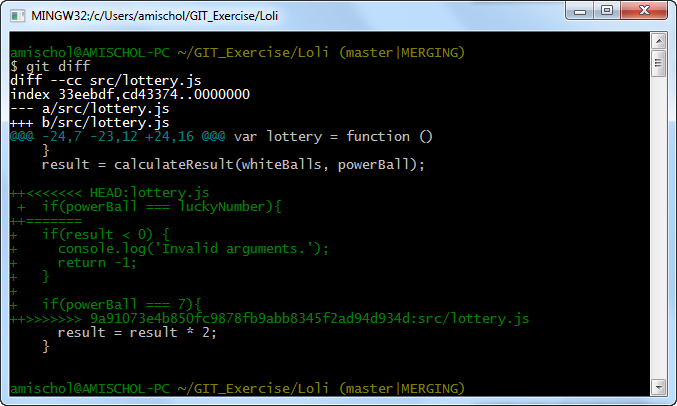


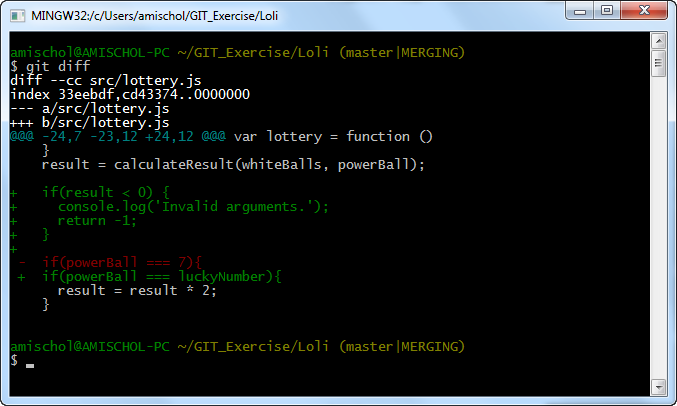
Loli needs to pull and merge before she can push her changes.



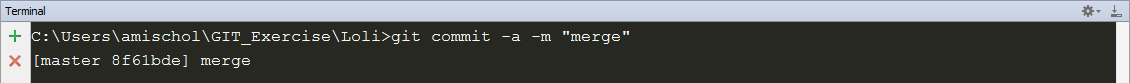
Let’s see the conflict

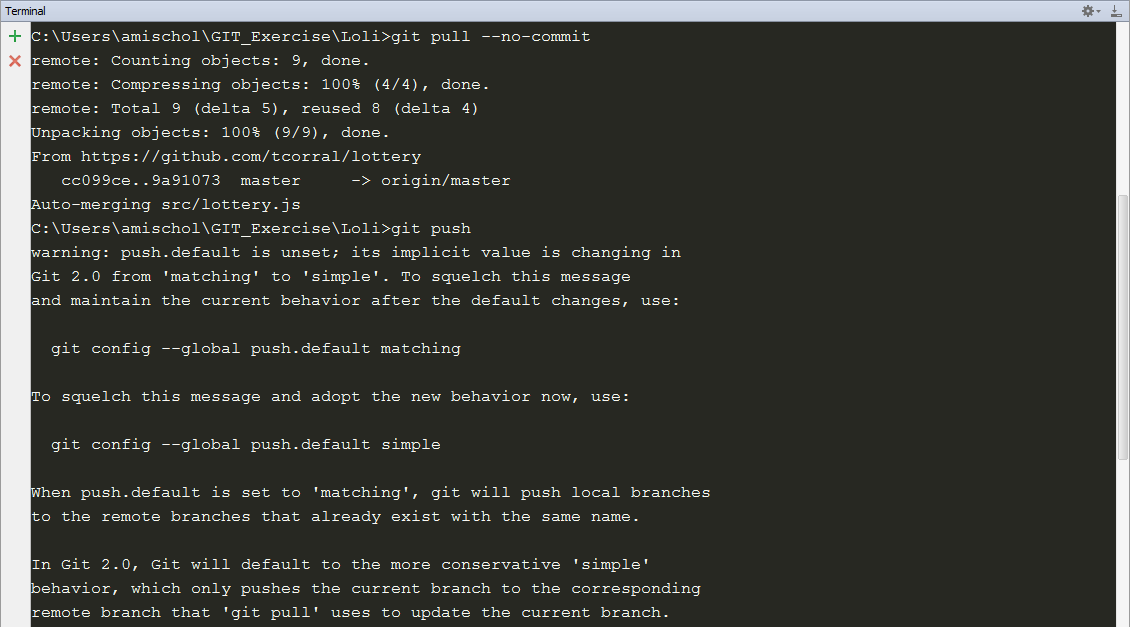


The change is easy so Loli fixes it.

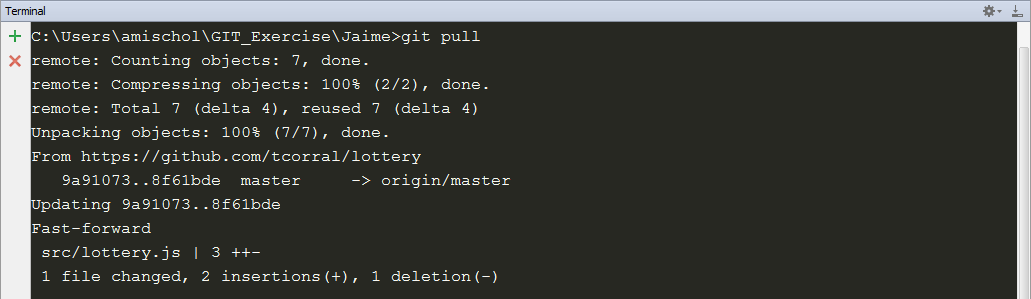


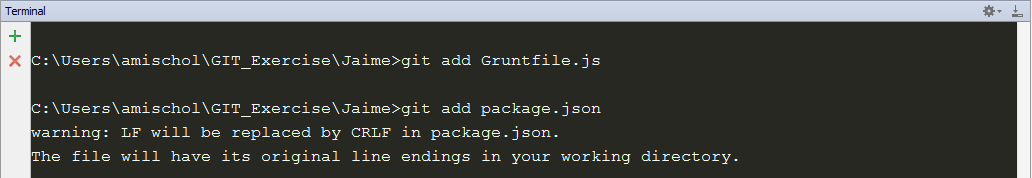
And commits and push the change.

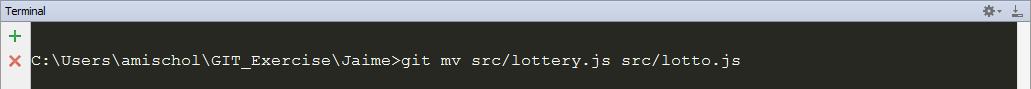


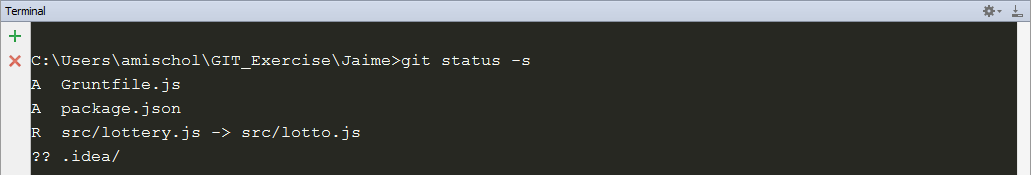
 

Jaime decides the time has come to create a proper Gruntfile.js and a package.json to lint our code with jshint and also rename lottery.js to lotto.js

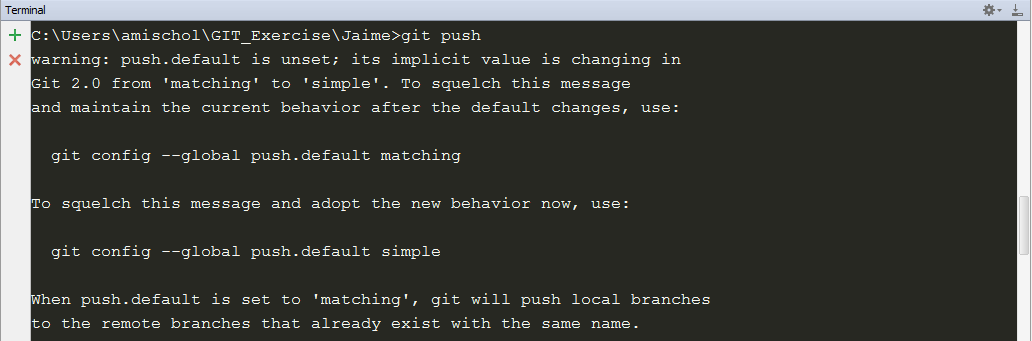


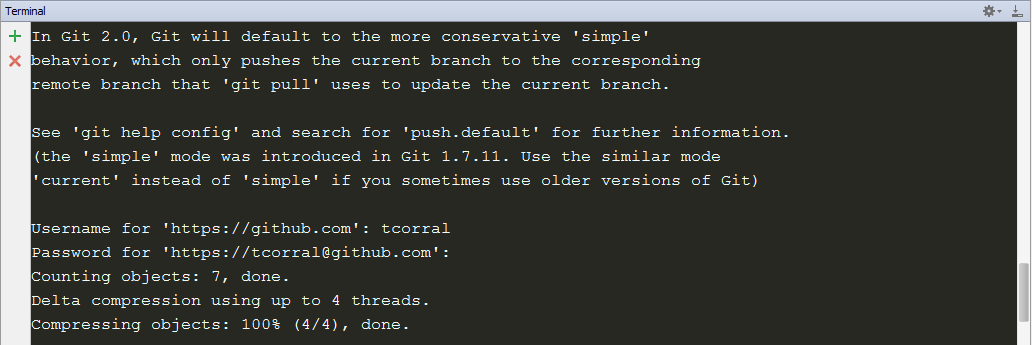
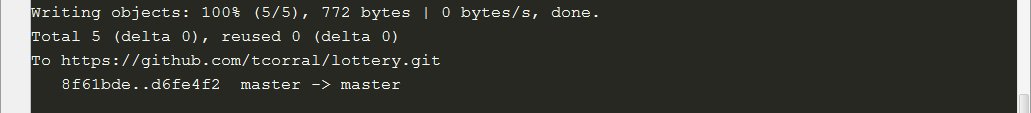




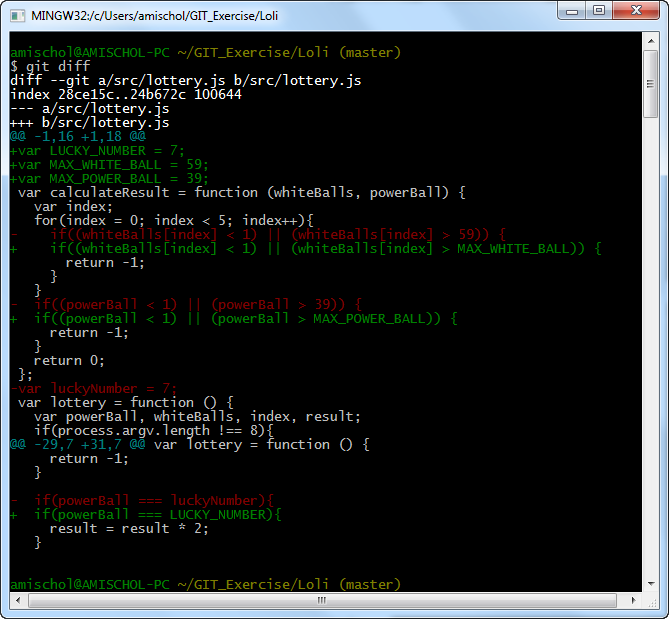




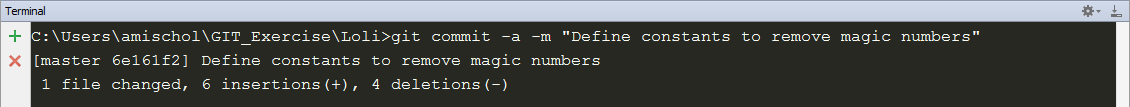


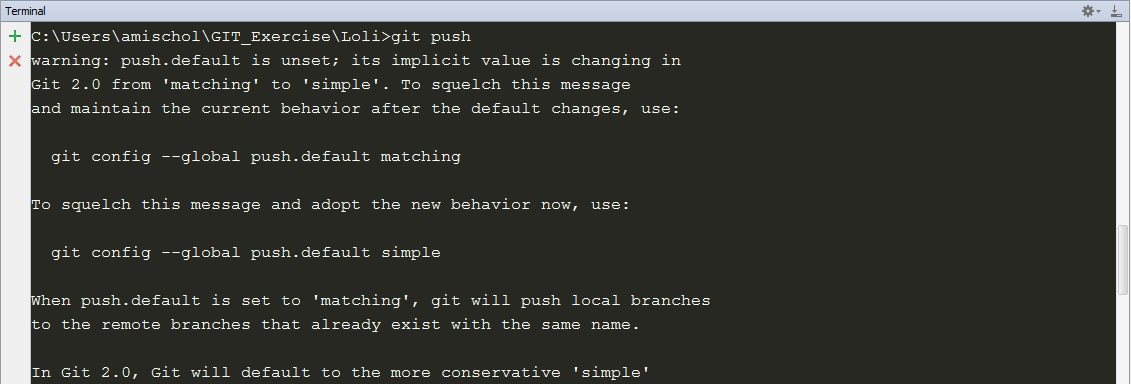
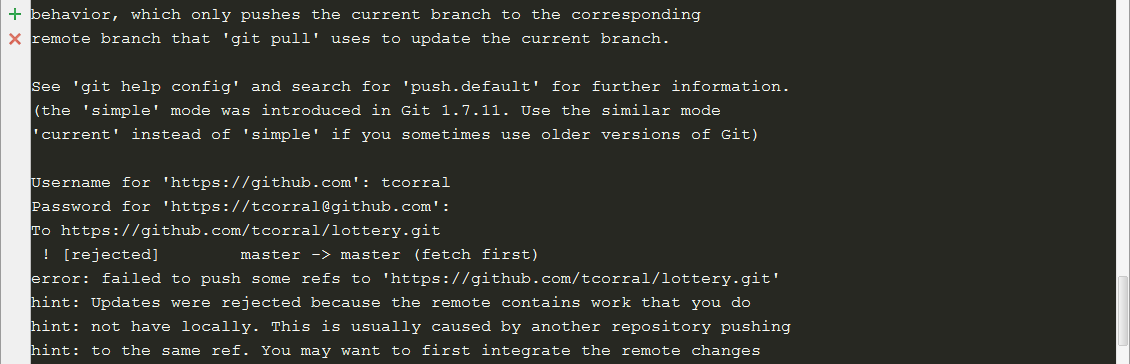
 

Meanwhile Loli changes luckyNumber to emulate constant and includes other magic numbers as max white ball value and max power ball value.

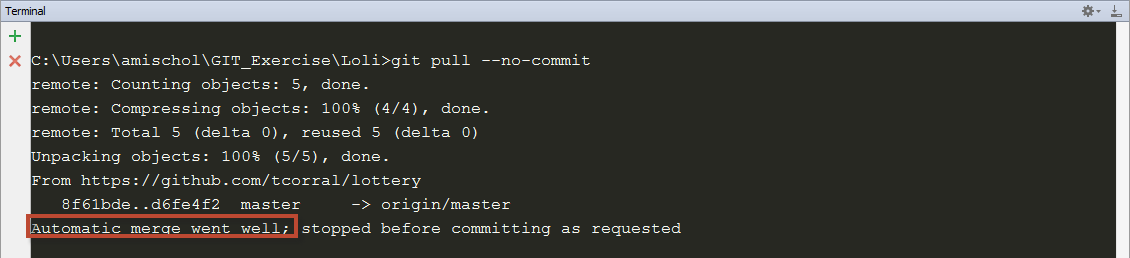


And commits her changes.

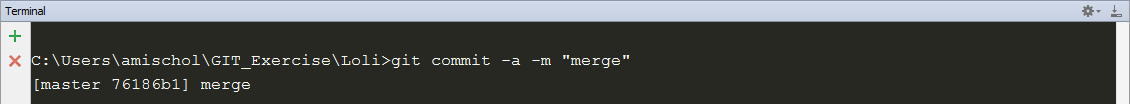


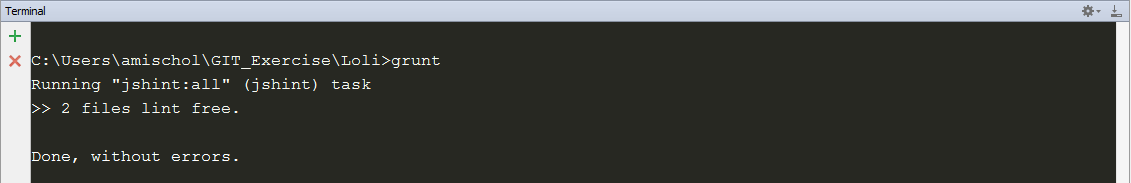
Again Jaime has pushed some change so Loli needs to pull changes again.

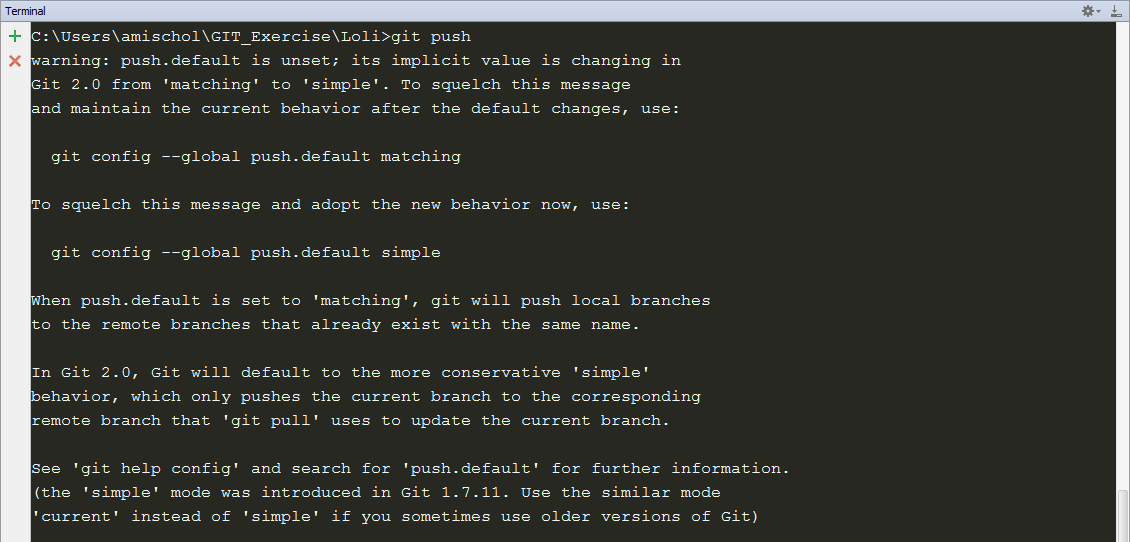
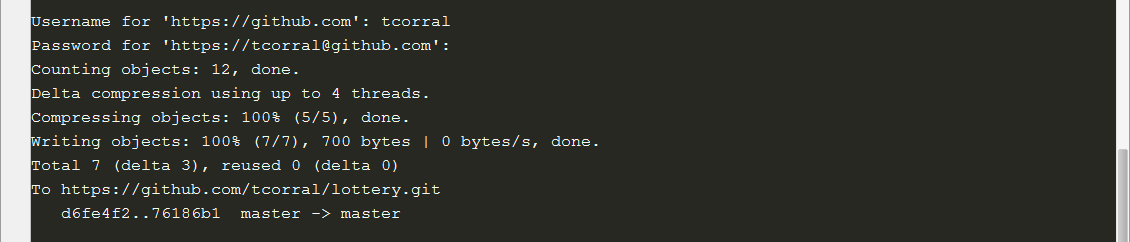


Loli can commit her changes without any problem.

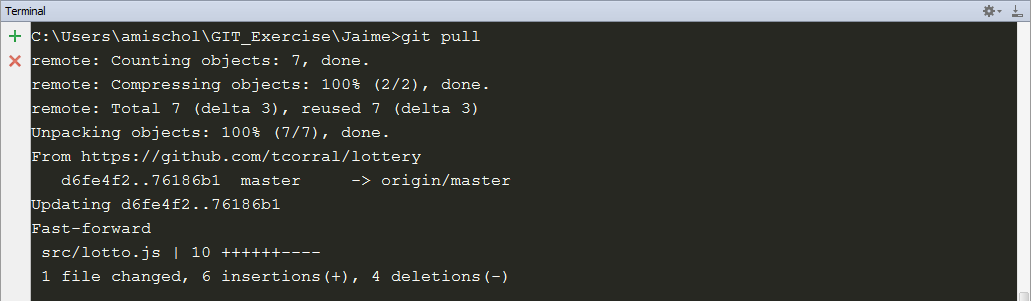


Now that we have a Gruntfile.js file we can launch it to check the code before pushing it.

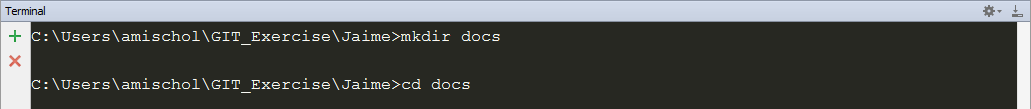


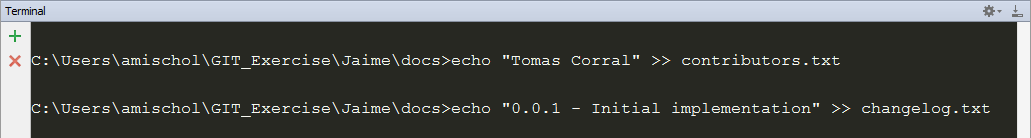
Jaime wants to to add the contributors and changelog files, so before doing any change he gets the last code in repo.



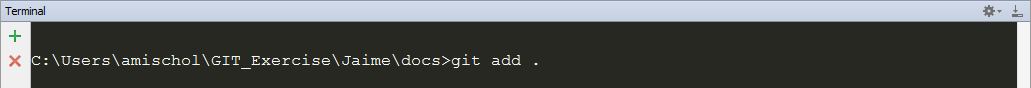
Jaime creates a docs folder where store the new files.



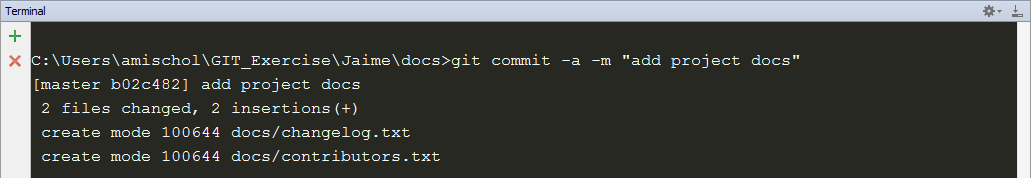
Create the two new files in docs folder

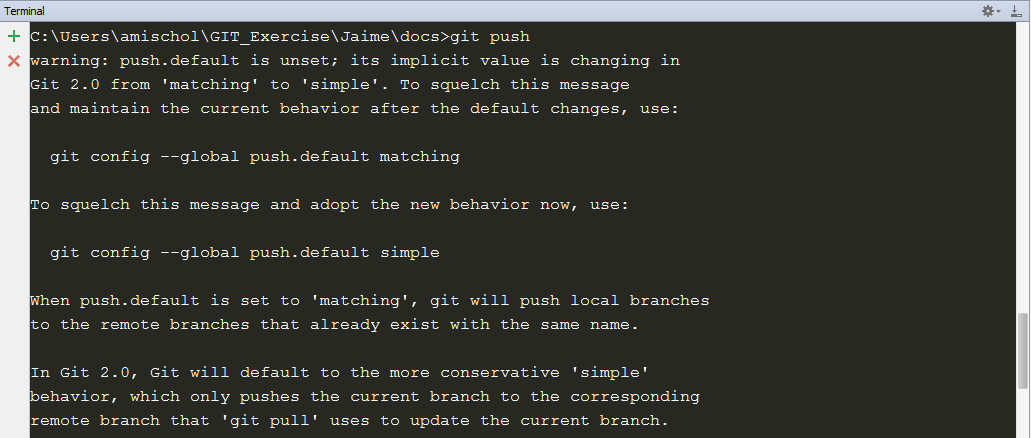


Add all the files inside docs folder

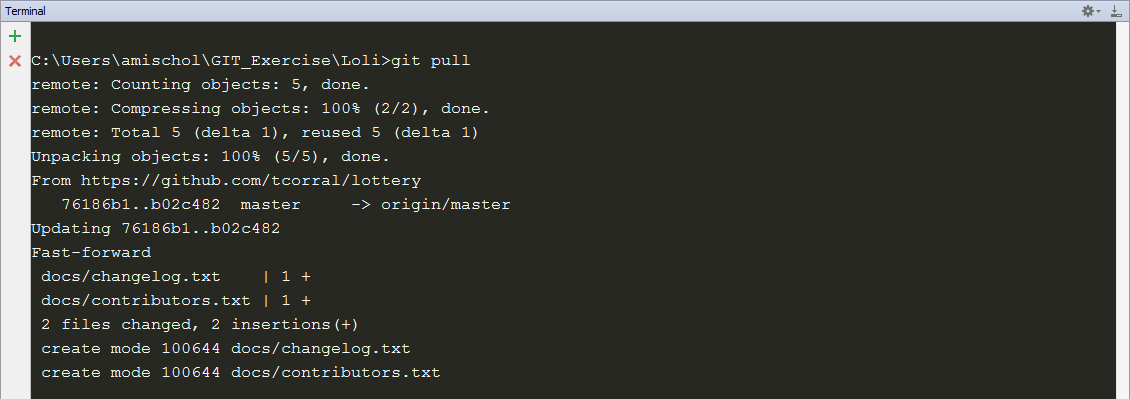


Commit and push the files.

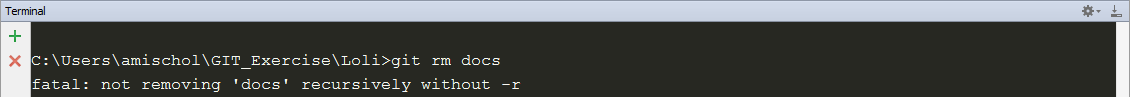


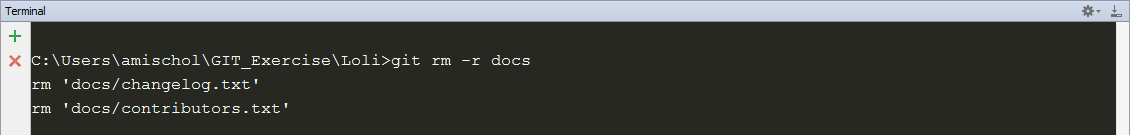
Loli pulls the changes from Jaime and see that these files are not a requirement from customer…



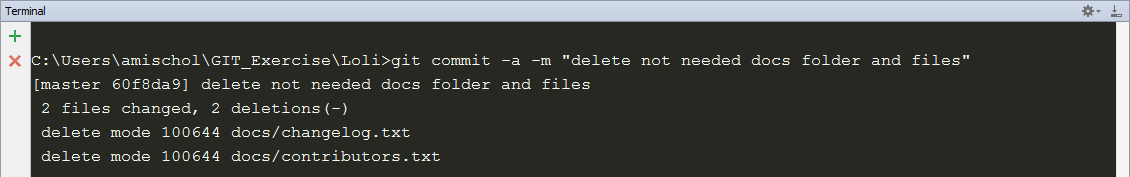
...so she decides to delete those files

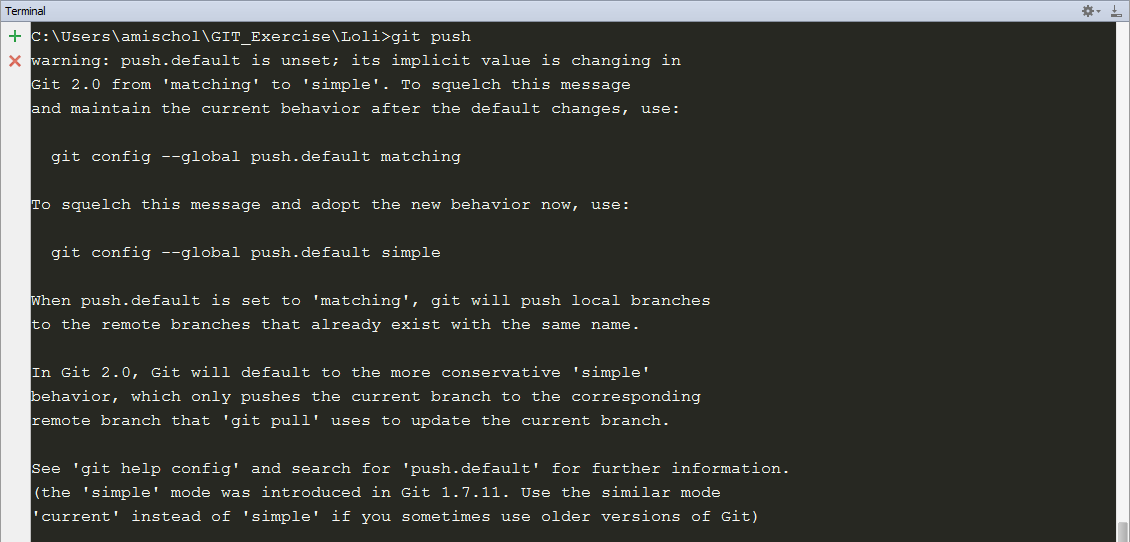
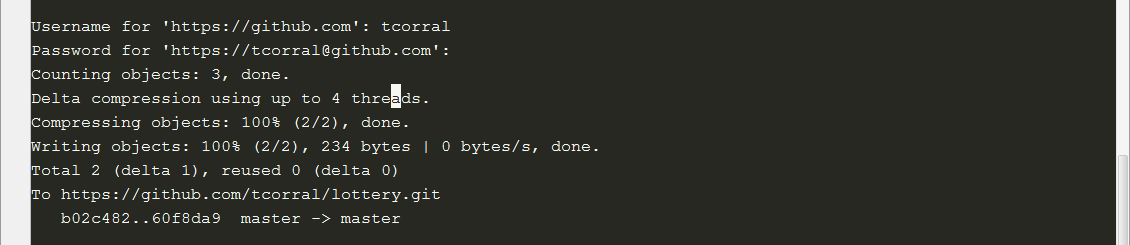


It needs to be recursive because is a folder with more than one file.

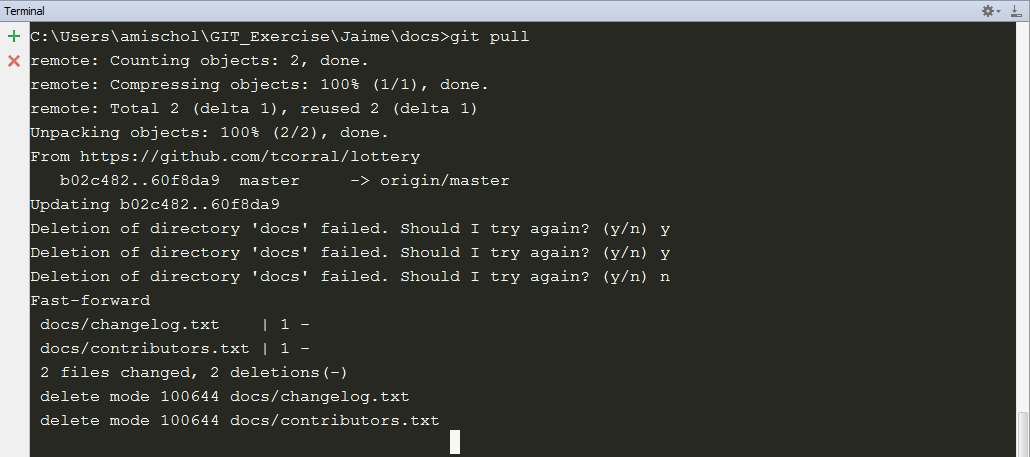


After being deleted the files it’s time to commit and push the changes.



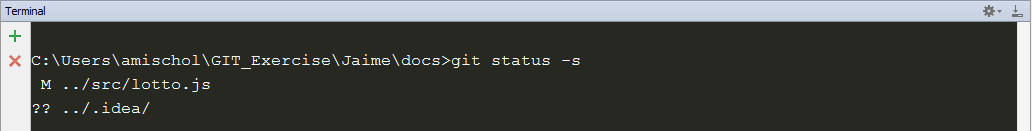
 

Jaime updates his repository instance

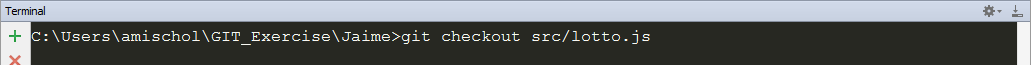


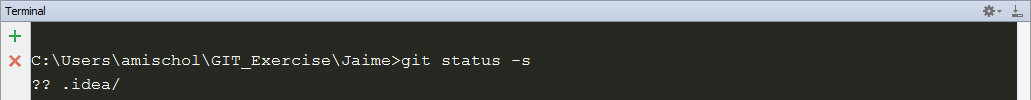
What’s wrong with Loli why she has decided to remove my docs folder? What if I changes the indentation of lotto.js just to be annoying?

Jaime changed the indentation of lotto.js from 2 spaces to 4 spaces.

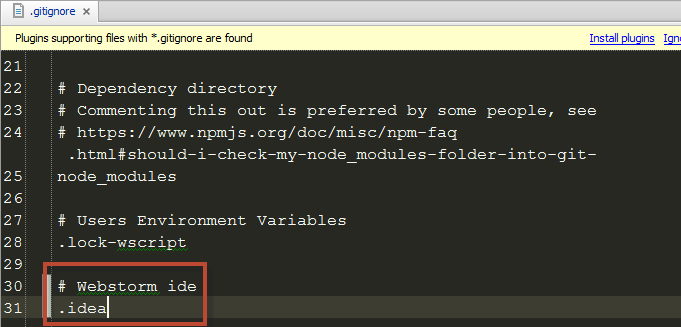


No, this change will be annoying not only for Loli, so Jaime dediced to revert his indentation changes.

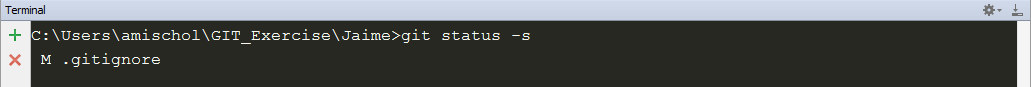




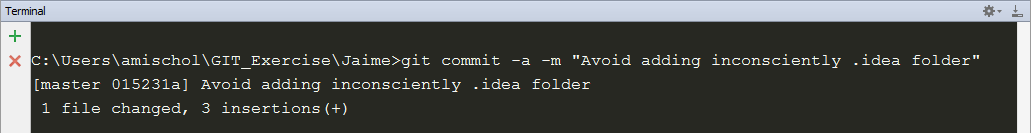
idea is shown because I use Webstorm so to avoid it to be expected to be added I added .idea in my .gitignore file.

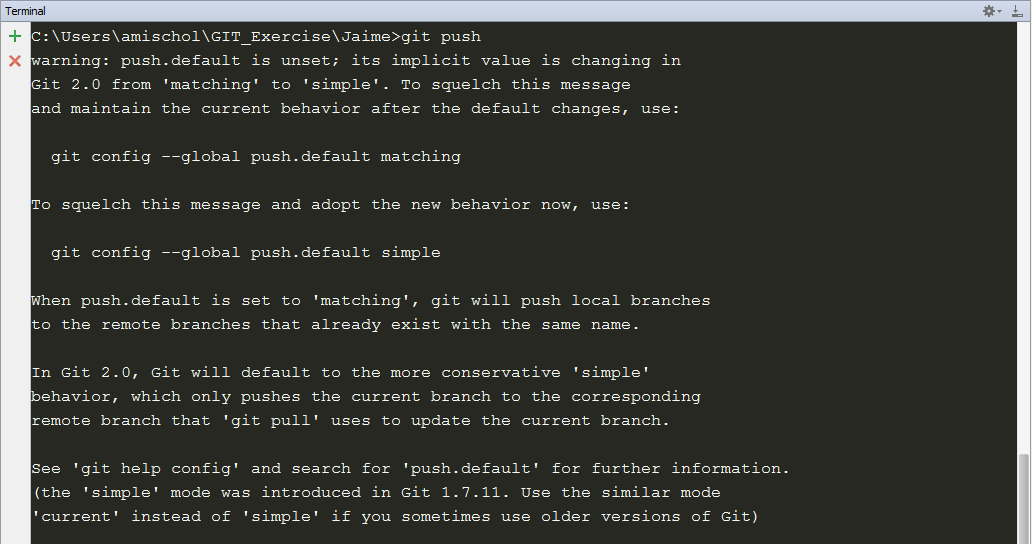


Jaime checks the status of repo.



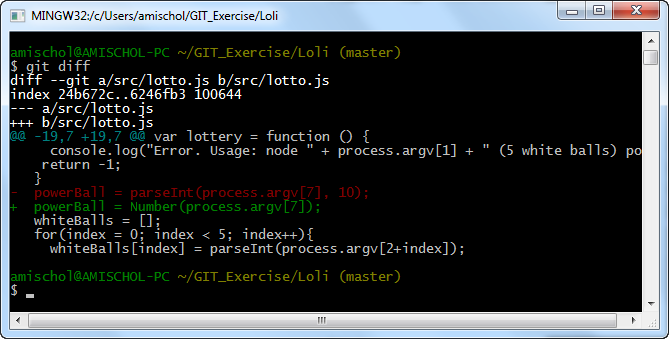
Commit and push this change.

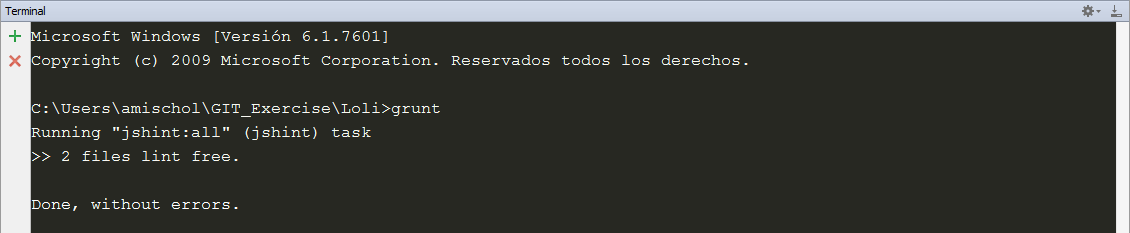


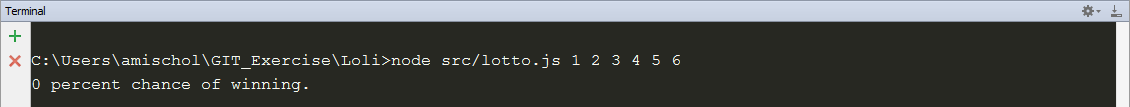


Loli made some performance tests and detected that cast String to Number is faster than using parseInt so she decides to change it.

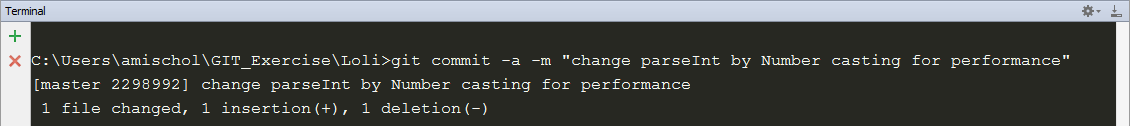


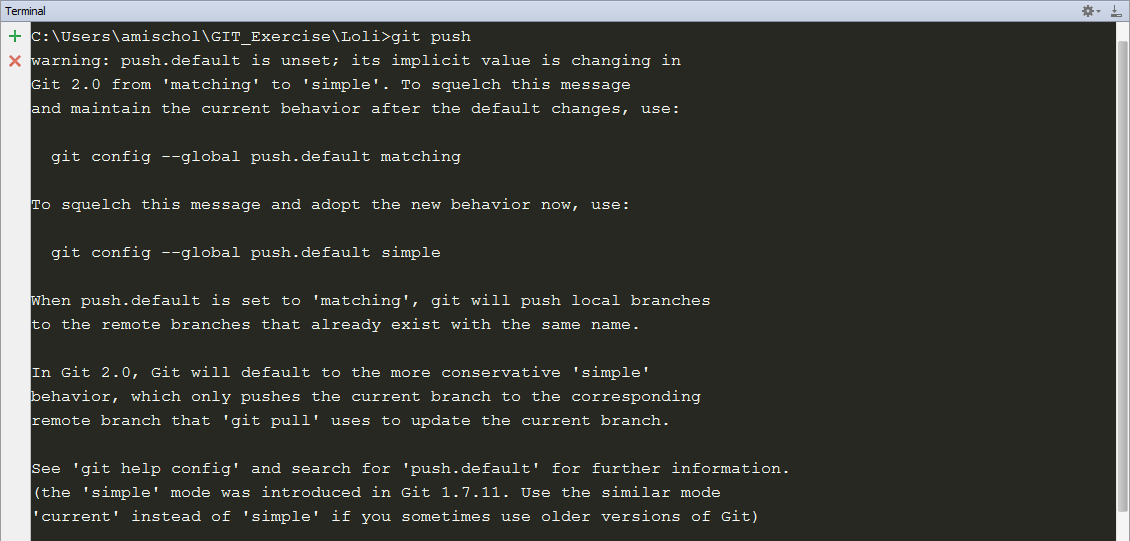


Loli checks that the code still works.

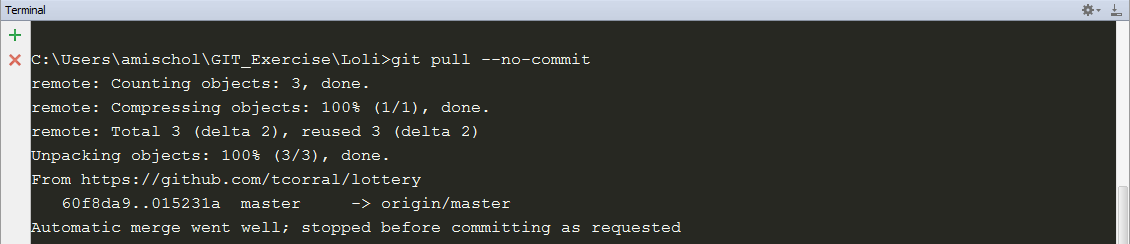


And commits and push her changes.



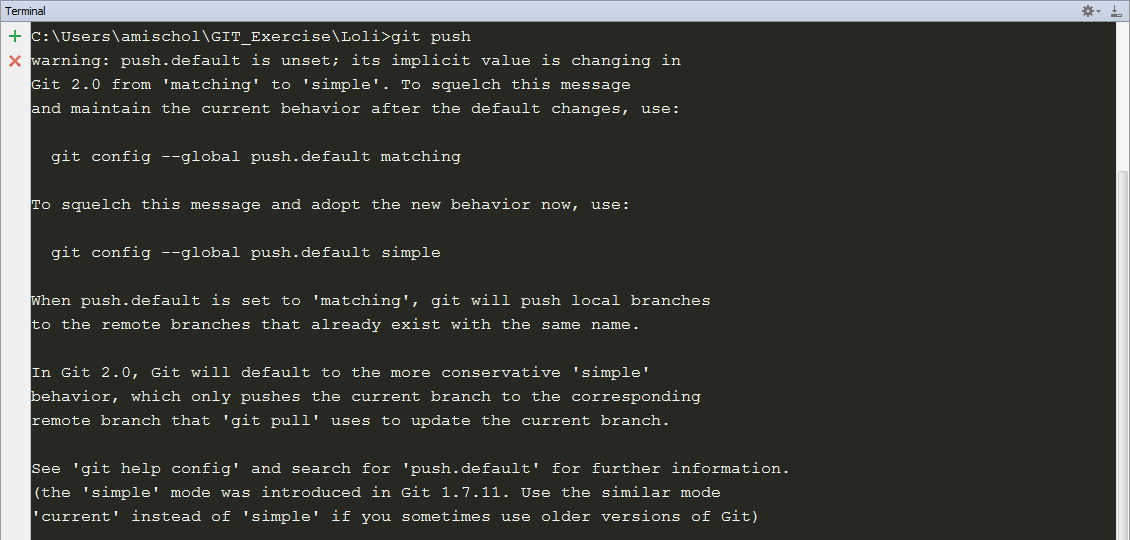
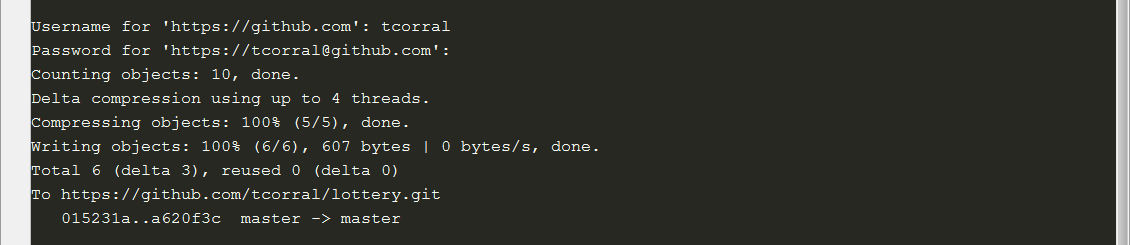
 

Umm! Something goes wrong, so it’s time to pull changes from Jaime.



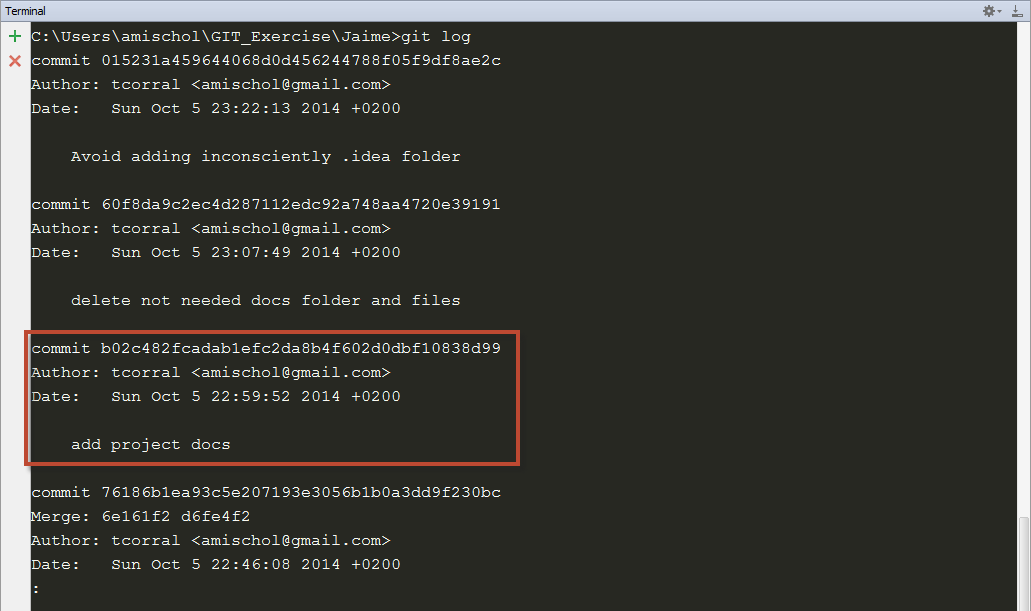
As we can see there is no problem to merge the changes so we commit and push the merge.

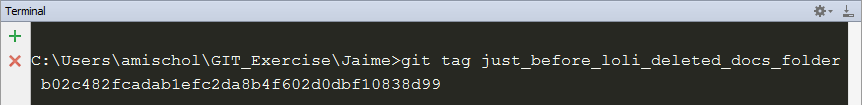


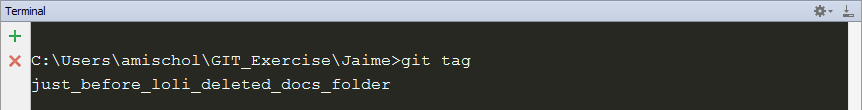
Jaime stills is disgusted by the deletion of the docs folder so he creates a tag so he can more easily access it later.

He search the logs for his commit to tag it before the deletion of doc folder.

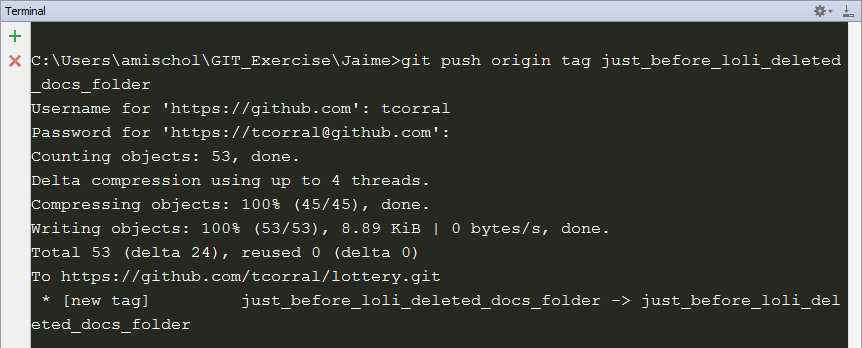




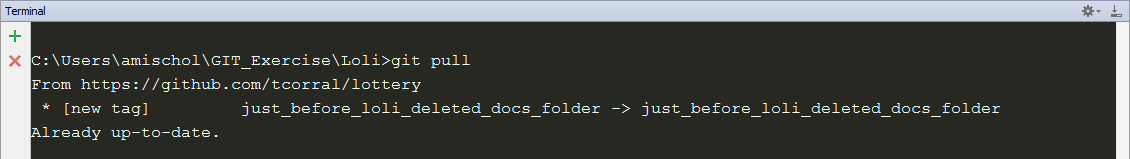
List the tags



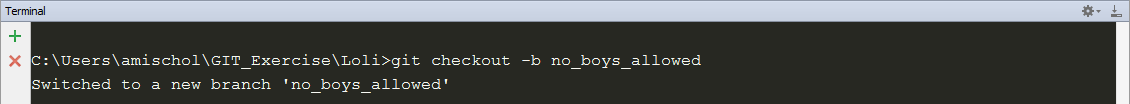
Jaime wants to share this tag, so he pushes the tag.



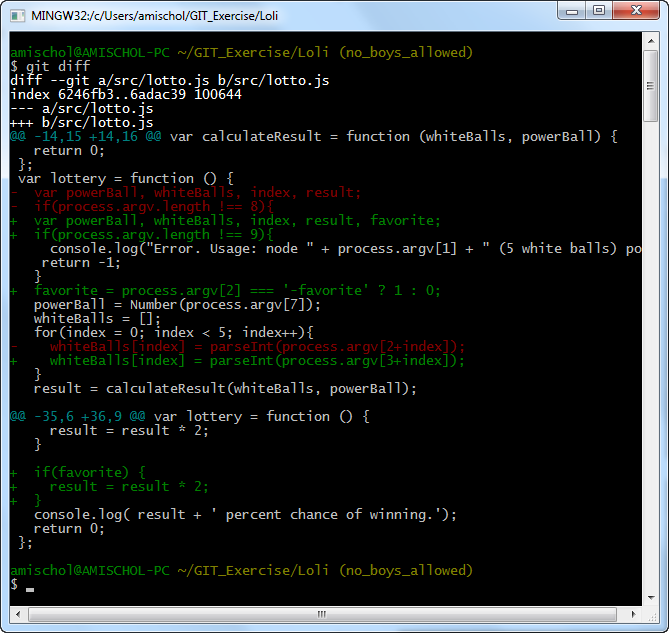
Jaime tells Loli about the new tag he created so she wants to see what he did.



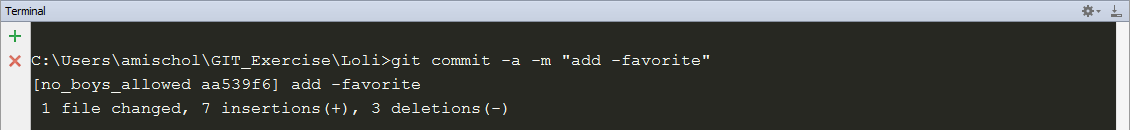
Loli wants to continue working but needs more privacy so she creates a new branch.

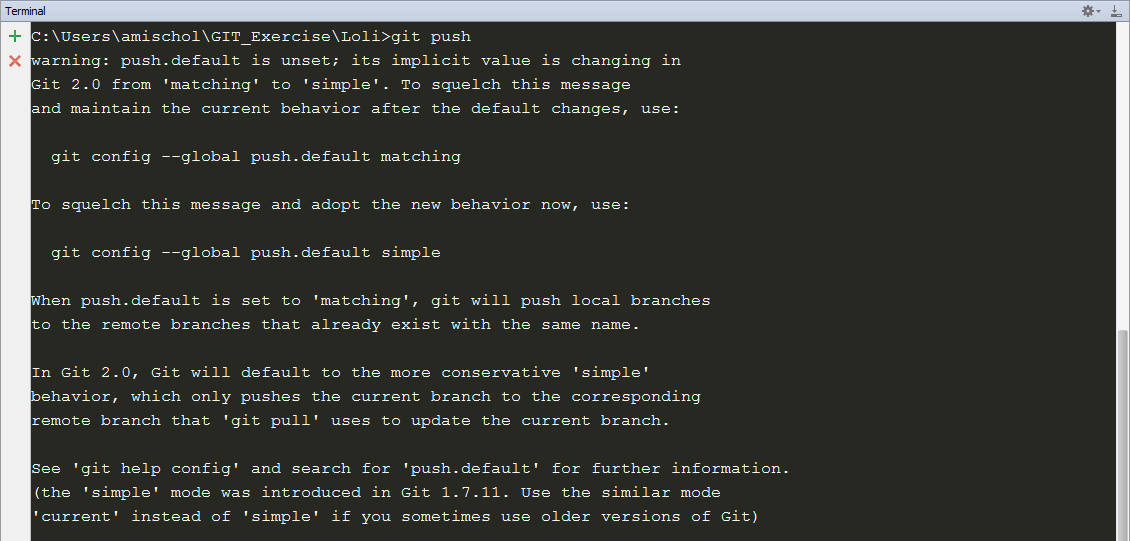
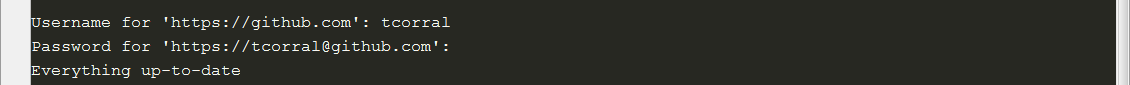


Now that can work without interferences she starts implementing a new feature that when a user plays her favorite numbers, her chances of winning should be doubled. She needs to rework the way command-line are parsed.

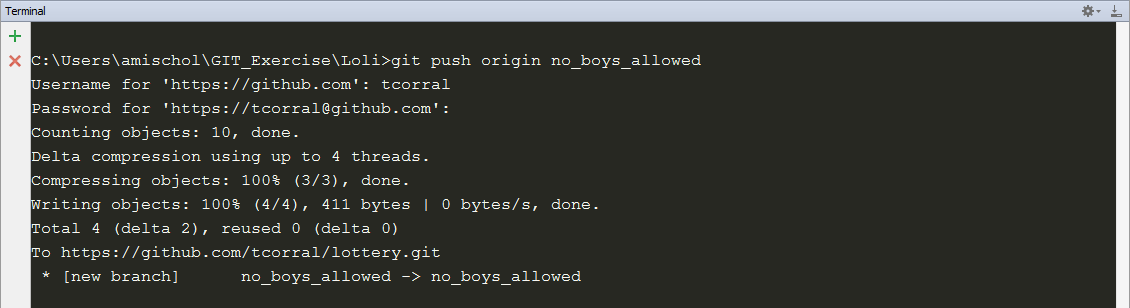


She commits her changes because she knows that no one will commit anything in this new branch.



What happened? Dammit Loli should be more explicit because now git has more that one branch so it needs Loli to tell git where to commit these changes.



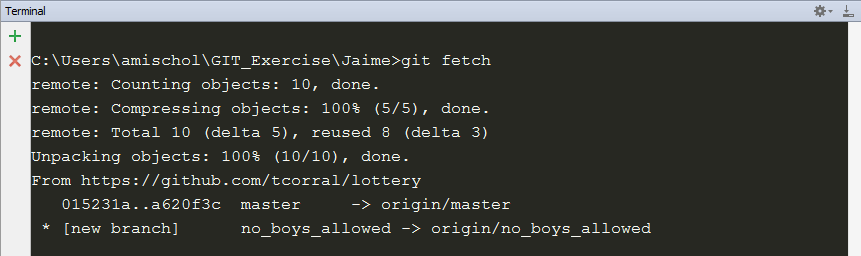
Meanwhile Jaime, in the master branch, decides that the white balls should be sorted before analyzing them.

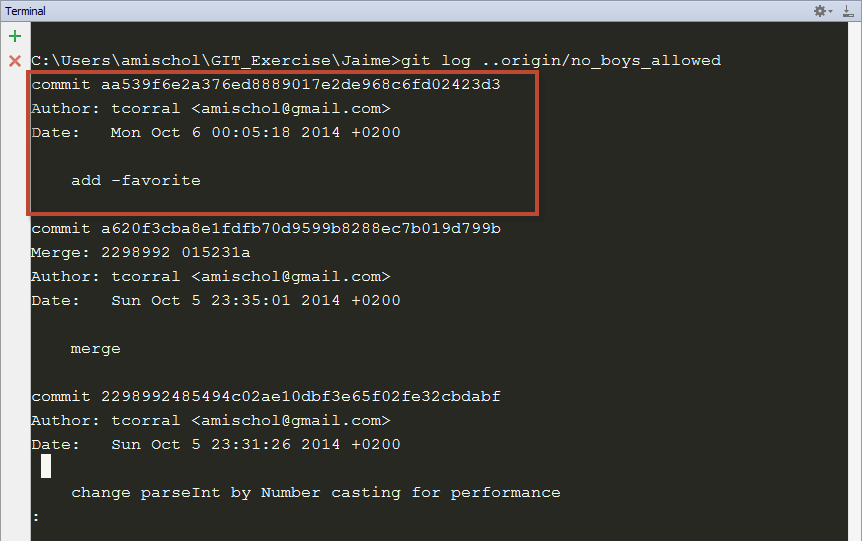


And commit his changes.

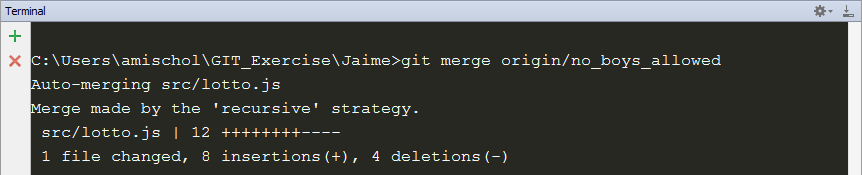


Before pushing his changes he wants to know what Loli have been working on, she said that he should not commit to her branch but she didn’t say anything about looking at it. ☺

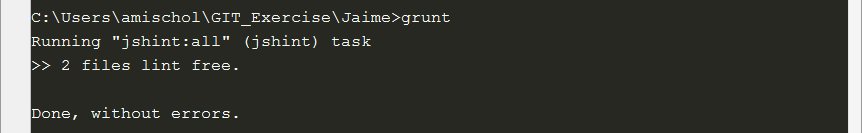




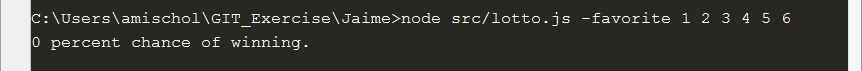
Jaime things that the favorite feature is very interesting and want to merge stuff from Loli branch into the default branch.



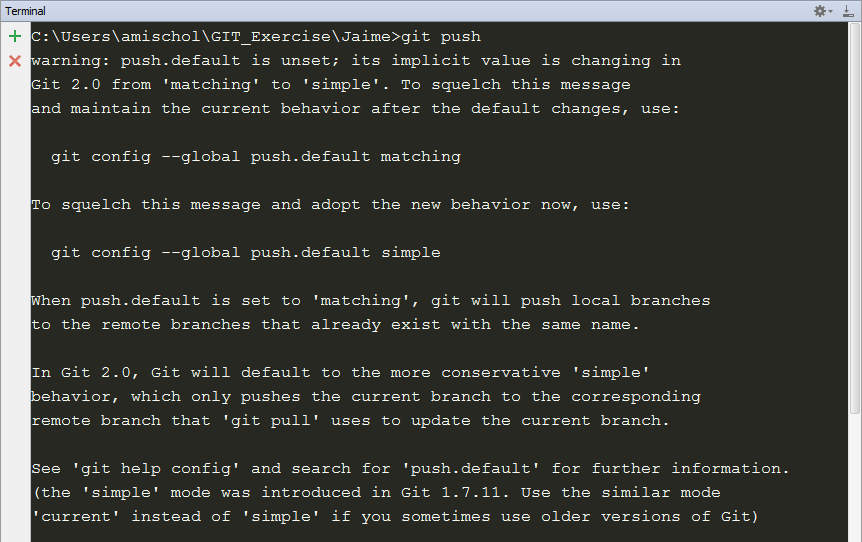
Jaime executes grunt to check that linting the code doesn’t raise any problem.



And test the new favorite feature.



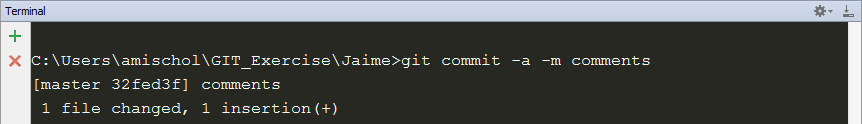
Whoo! It works. Jaime decides to push this changes to the default branch.

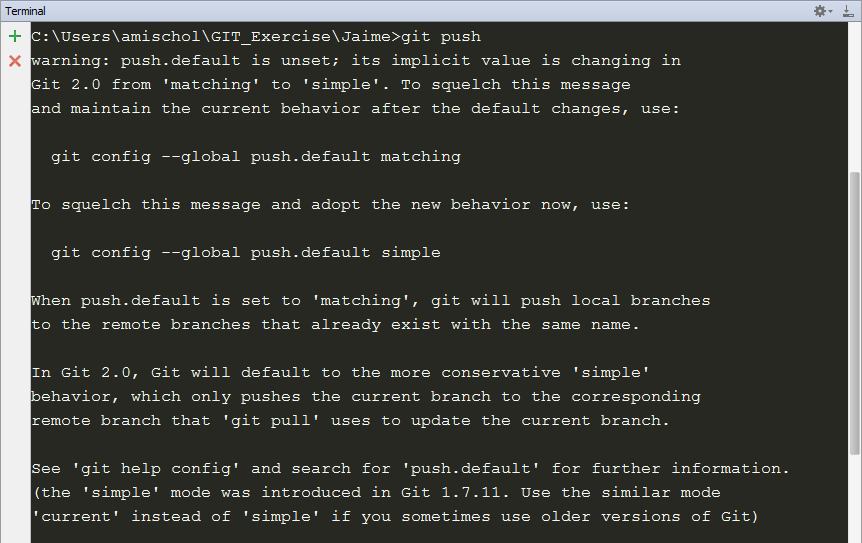
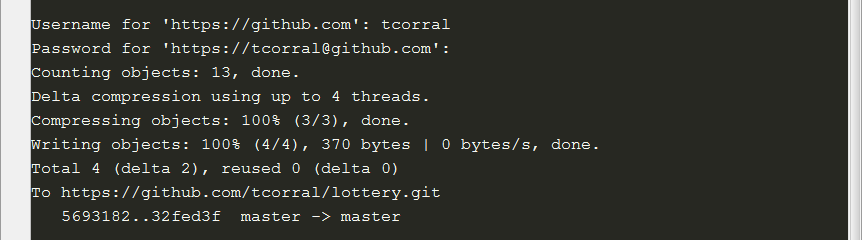
 

Oh, no! Where are the comments in code? Jaime and Loli realize that their code has no comments and it could be a great problem if they have to revisit their code.

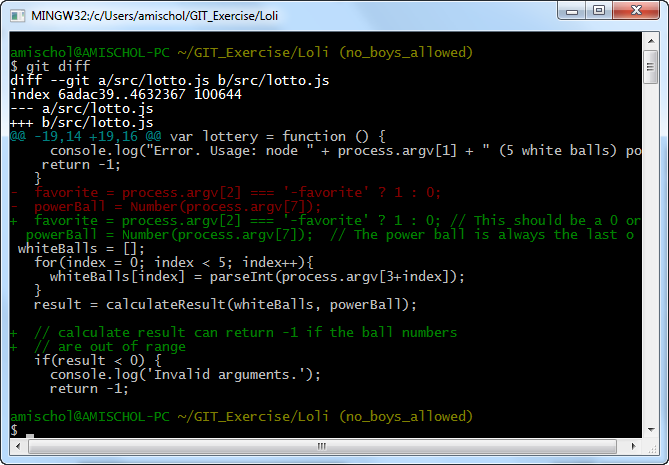
Jaime’s changes:

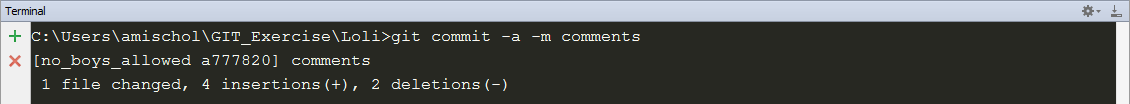


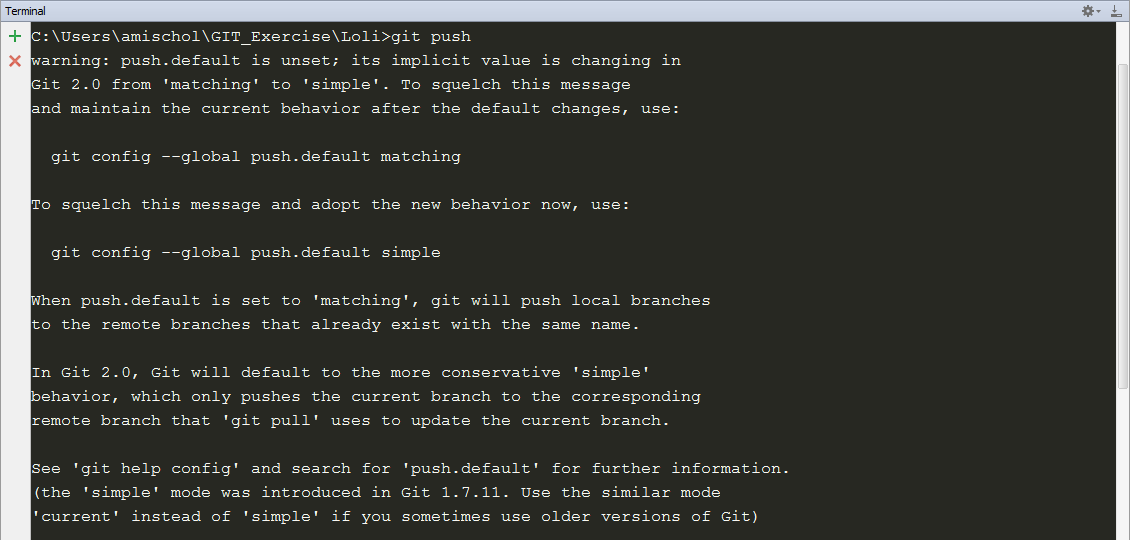
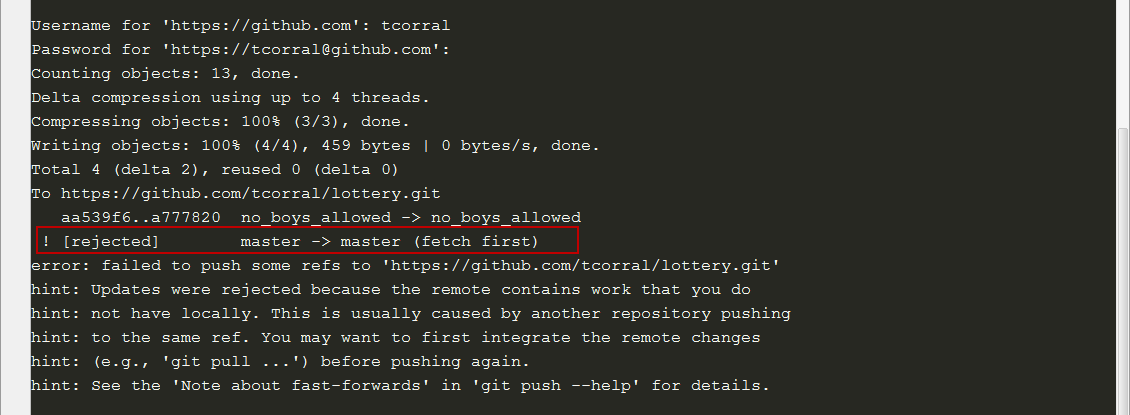


Loli’s changes:



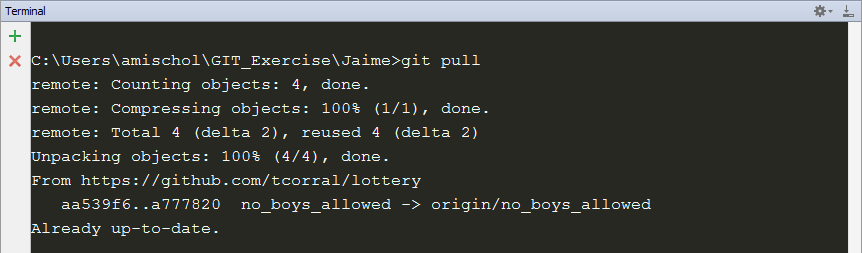


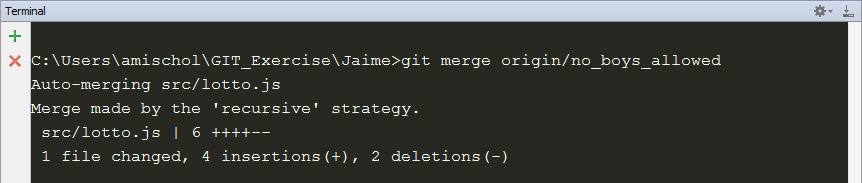
 

Loli notices that the push of her private suceeded. Git seems to be complaining about something related to the master branch. She thinks it best that she just ignore it.

*(This error is saying Loli that her repository instance is out of date)*

Jaime tries to merge the changes from Loli’s branch.

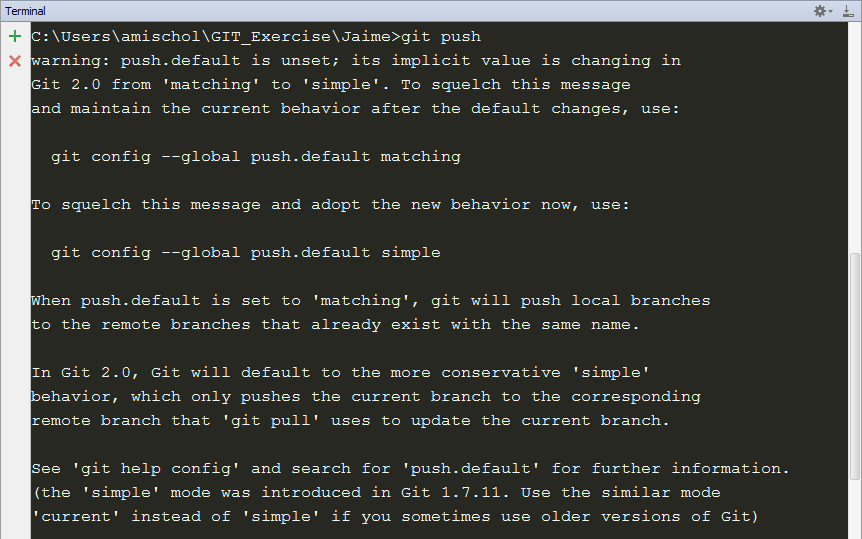




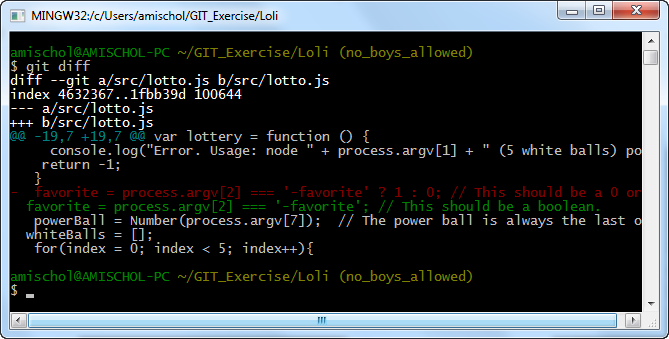
No problem merging the changes, but to be sure Jaime executes the grunt default task to lint the code.



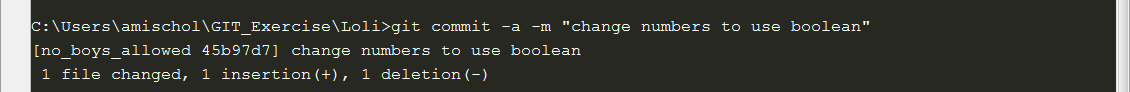
All seems fine so it push the changes of the merge.

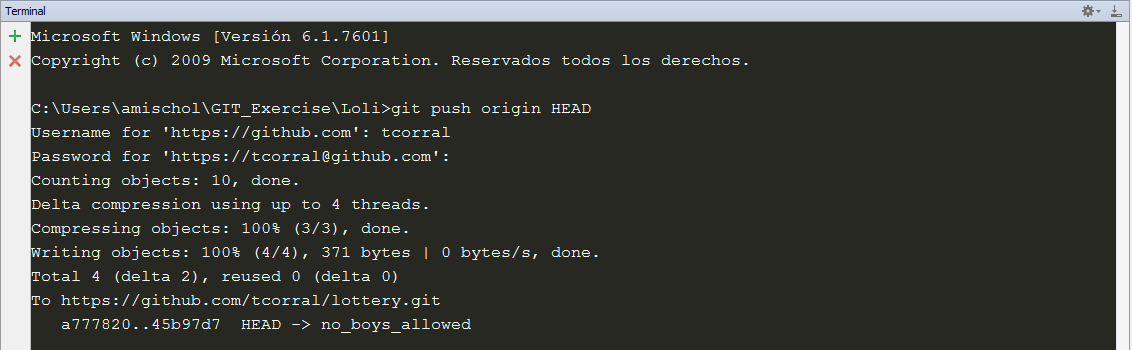
 

Loli realized that in the code they are using numbers instead of using Boolean values so she decides to fix it.

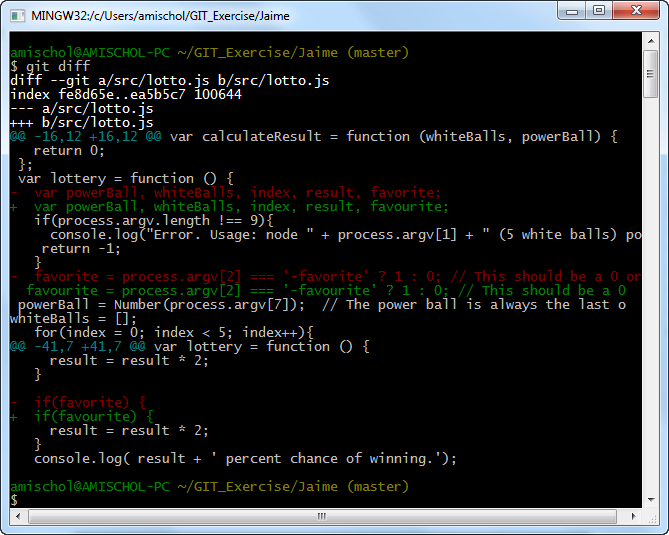


And she commits the change to her private branch.

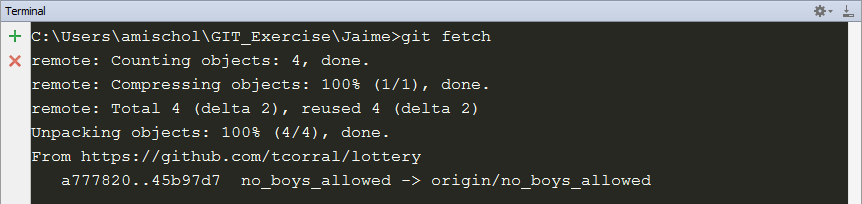


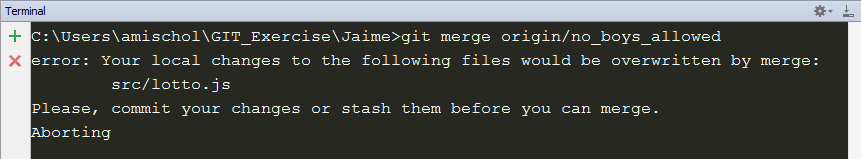


Meanwhile Jaime that has a lot of knowledge about grammar he wants to correct the spelling of the word “favourite”.

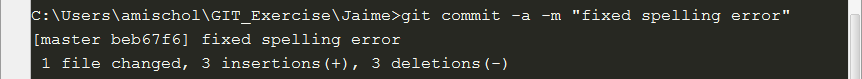


Jaime is proud about his change so it proceeds to commit the change and to once again merge Sally’s changes into the default branch.

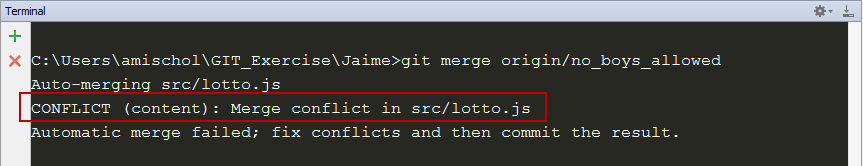




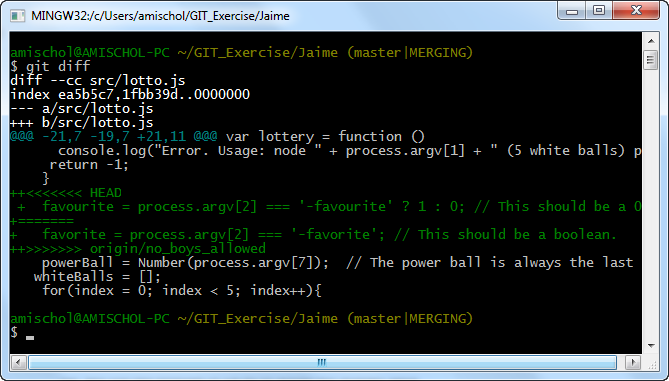
Oh, maybe Jaime forgot to commit his changes?



Yes, this was the problem so we try to merge the changes again.



Dammit, Jaime has merge conflicts with lotto.js file so he checks what the problem is.



Jaime realizes that this conflict needs to be resolved manually by keeping the proper spelling but converting the type to Boolean like Loli did.

