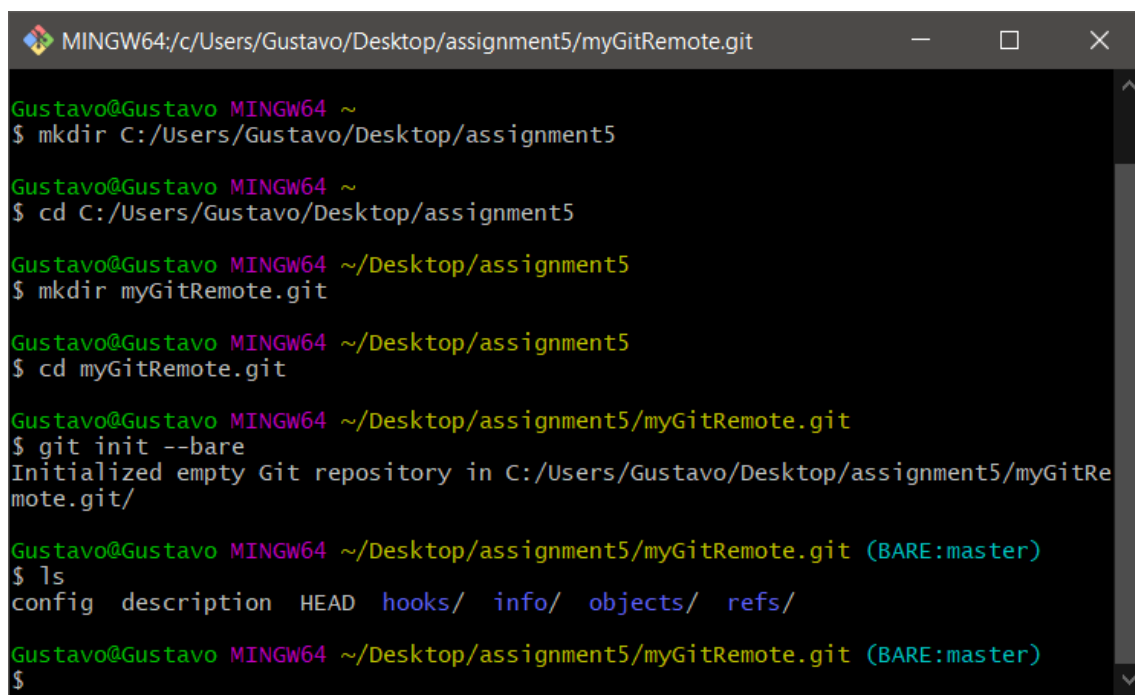


Name: Gustavo Hammerschmidt.

## Screenshot 1:



```
MINGW64:/c/Users/Gustavo/Desktop/assignment5/myGitRemote.git
Gustavo@Gustavo MINGW64 ~
$ mkdir C:/Users/Gustavo/Desktop/assignment5

Gustavo@Gustavo MINGW64 ~
$ cd C:/Users/Gustavo/Desktop/assignment5

Gustavo@Gustavo MINGW64 ~/Desktop/assignment5
$ mkdir myGitRemote.git

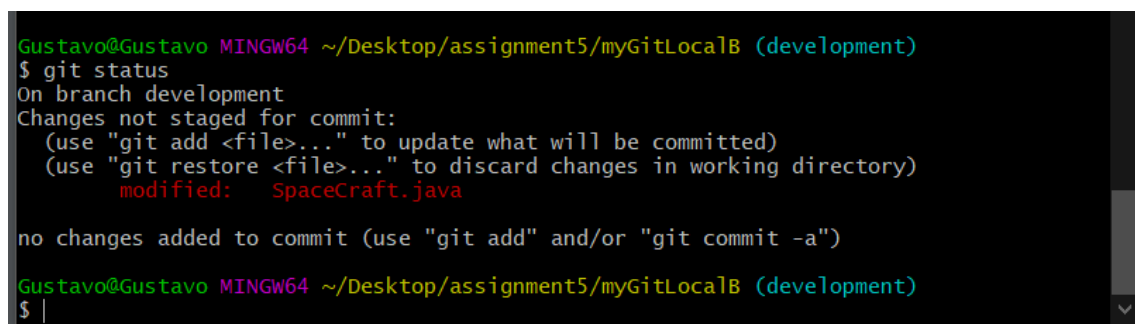
Gustavo@Gustavo MINGW64 ~/Desktop/assignment5
$ cd myGitRemote.git

Gustavo@Gustavo MINGW64 ~/Desktop/assignment5/myGitRemote.git
$ git init --bare
Initialized empty Git repository in C:/Users/Gustavo/Desktop/assignment5/myGitRemote.git/

Gustavo@Gustavo MINGW64 ~/Desktop/assignment5/myGitRemote.git (BARE:master)
$ ls
config  description  HEAD  hooks/  info/  objects/  refs/

Gustavo@Gustavo MINGW64 ~/Desktop/assignment5/myGitRemote.git (BARE:master)
$
```

## Screenshot 2:



```
Gustavo@Gustavo MINGW64 ~/Desktop/assignment5/myGitLocalB (development)
$ git status
On branch development
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   SpaceCraft.java

no changes added to commit (use "git add" and/or "git commit -a")

Gustavo@Gustavo MINGW64 ~/Desktop/assignment5/myGitLocalB (development)
$ |
```

### Screenshot 3 (Merge conflict message):

```
Gustavo@Gustavo MINGW64 ~/Desktop/assignment5/scenarioB (master|MERGING)
$ git merge origin/scenarioB_branch
error: Merging is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.

Gustavo@Gustavo MINGW64 ~/Desktop/assignment5/scenarioB (master|MERGING)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)

You have unmerged paths.
(fix conflicts and run "git commit")
(use "git merge --abort" to abort the merge)

Unmerged paths:
(use "git add <file>..." to mark resolution)
    both modified:   GameController.java

no changes added to commit (use "git add" and/or "git commit -a")
```

### Screenshot 3 (Merge diff view of changes on both files):

```

-import java.io.BufferedReader;
import java.io.IOException;
+import java.io.BufferedReader;
import java.io.InputStreamReader;

+// I've imported IOException first.
+

public class GameController {

@@ -20,8 +22,11 @@ public class GameController {

    public void run(){
        //Send welcome message
-        System.out.println("#Welcome to Lunar Lander");
+        //System.out.println("#Welcome to Lunar Lander");

+        System.out.println("This is Lunar Lander");
+        // I've changed the title displayed.
+

        try{
            //Begin reading from System input
            BufferedReader inputReader =

commit 7981a7ce427e0ceb29314543a78c4c5da5276bd3 (HEAD -> master, origin/scenarioA_branch)
Author: Gustavo <gustavocrazy@yahoo.com>
Date: Thu May 7 00:09:23 2020 -0300

    scenarioA_changes

diff --git a/GameController.java b/GameController.java
index 4f335f6..039790d 100644
--- a/GameController.java
+++ b/GameController.java
@@ -1,6 +1,8 @@
+import java.io.InputStreamReader;
import java.io.BufferedReader;
import java.io.IOException;
-import java.io.InputStreamReader;
+
+// I've imported InputStreamReader first.

public class GameController {

@@ -20,8 +22,14 @@ public class GameController {

    public void run(){
        //Send welcome message
-        System.out.println("#Welcome to Lunar Lander");
+        //System.out.println("#welcome to Lunar Lander");

+        System.out.print("\n");
+        System.out.print("Welcome to my Lunar Lander app");
+        System.out.print("\n");
+
+        // I've changed the introduction message;
+

        try{

```

Screenshot 3 (Merge conflict fixed):

```

Gustavo@Gustavo MINGW64 ~/Desktop/assignment5
$ cd scenarioA

Gustavo@Gustavo MINGW64 ~/Desktop/assignment5/scenarioA (developmentA)
$ git remote origin
error: Unknown subcommand: origin
usage: git remote [-v | --verbose]
    or: git remote add [-t <branch>] [-m <master>] [-f] [--tags | --no-tags] [
--mirror=<fetch|push>] <name> <url>
    or: git remote rename <old> <new>
    or: git remote remove <name>
    or: git remote set-head <name> (-a | --auto | -d | --delete | <branch>)
    or: git remote [-v | --verbose] show [-n] <name>
    or: git remote prune [-n | --dry-run] <name>
    or: git remote [-v | --verbose] update [-p | --prune] [(<group> | <remote>
)...]
    or: git remote set-branches [--add] <name> <branch>...
    or: git remote get-url [--push] [--all] <name>
    or: git remote set-url [--push] <name> <newurl> [<oldurl>]
    or: git remote set-url --add <name> <newurl>
    or: git remote set-url --delete <name> <url>

    -v, --verbose          be verbose; must be placed before a subcommand

Gustavo@Gustavo MINGW64 ~/Desktop/assignment5/scenarioA (developmentA)
$ git fetch origin

Gustavo@Gustavo MINGW64 ~/Desktop/assignment5/scenarioA (developmentA)
$ git checkout -b scenariob_development origin/scenarioA_branch
Switched to a new branch 'scenariob_development'
Branch 'scenariob_development' set up to track remote branch 'scenarioA_branc
h' from 'origin'.

```

Question1:

No, the changes I have made did not show up because the development branch was not merged to the master branch.

Question 2:

Yes, because I have fetched the latest data from the repository.

## Description of Task 1:

- I have made two local repositories, named: scenarioA and scenarioB. On both, I have changed the order of the imports in a way both would have the same imports, but in a different calling order. I have made changes to the way the title: "Welcome to Lunar Lander" was displayed in both files. To solve the merging conflict, I have executed the command diff to check where the conflicts happened, and then picked one of both files and manually changed the file to avoid conflicting with the other scenario file. Then, I have fetched the scenario with the origin and both scenarios had the changes I chose up to date. I sent three screenshots of number 3. screenshot3\_merge\_conflict\_message shows the first time that a merge conflict message was reported; screenshot3\_merge\_diff\_view shows the diff view of the changes on both files and screenshot3\_merge\_conflict\_fixed shows the message that the conflict was resolved. I have sent a file with all the commands executed on the prompt throughout the whole assignment, named:

prompt\_command\_execution.txt. I have also sent a zip folder that contains the git repository, screenshots, this file on docx version and the text files that I have made when working on the assignment as notes.