

## Assignment 4

### Restoring a Deleted File in Git: BlackBoxGiven.java

#### **1. What would happen if we merged Whitebox into master after deleting BlackBoxGiven.java?**

If we merged the Whitebox branch (where BlackBoxGiven.java was deleted) into the master branch, the file would also be deleted from master. This is not what we want, since we want to keep our test files. This demonstrates how changes in one branch (like deletions) can propagate to other branches during merges, potentially causing unwanted data loss.

#### **2. Steps to Restore BlackBoxGiven.java in StaticAnalysis Branch**

##### **a. Switch to the Whitebox branch and create StaticAnalysis branch**

`git checkout Whitebox`

```
$ git checkout whitebox
Switched to branch 'whitebox'
```

`git checkout -b StaticAnalysis`

```
$ git checkout -b StaticAnalysis
Switched to a new branch 'StaticAnalysis'
```

##### **b. Find the commit where the file was deleted**

`git log --diff-filter=D --summary`

```
In (StaticAnalysis)
$ git log --diff-filter=D --summary
commit cab5195a9e9818c54c863b67355b2034685e4d52 (HEAD -> StaticAnalysis, Whitebox)

Delete BlackBoxGiven.java from Whitebox branch

delete mode 100644 src/test/java/BlackBoxGiven.java
```

**c. Restore the file from the commit before deletion**

```
git checkout Whitebox~1 src/test/java/BlackBoxGiven.java
```

```
$ git checkout Whitebox~1 src/test/java/BlackBoxGiven.java  
Updated 1 path from 52721bb
```

**d. Add and commit the restored file**

```
git add src/test/java/BlackBoxGiven.java
```

```
$ git add src/test/java/BlackBoxGiven.java
```

```
git commit -m "Restore BlackBoxGiven.java in StaticAnalysis branch"
```

```
$ git commit -m "Restore BlackBoxGiven.java in StaticAnalysis branch"  
[StaticAnalysis 0e81af] Restore BlackBoxGiven.java in StaticAnalysis branch  
1 file changed, 176 insertions(+)  
create mode 100644 src/test/java/BlackBoxGiven.java
```

**Not Running All Test Files**

**Steps Completed:**

1. Modified build.gradle to exclude BlackBoxGiven tests:

```
test {  
    exclude '**/BlackBoxGiven.class'  
}
```

2. Successfully ran gradle clean and gradle build on the Blackbox branch
  - The build was successful, showing that BlackBoxGiven tests were properly excluded

```
$ gradle build  
[Incubating] Problems report is available at: file:///C:/Users/ahdma/OneDrive/Documents/2nd%20CSE/Eyad/ser316-summer25C-eghanem-main/ser316-summer25C-eghanem-main/build/reports/problems/problems-report.html  
Deprecated Gradle features were used in this build, making it incompatible with Gradle 9.0.  
You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.  
For more on this, please refer to https://docs.gradle.org/8.12/userguide/command_line_interface.html#sec:command_line_warnings  
BUILD SUCCESSFUL in 2s  
8 actionable tasks: 8 up-to-date
```

### 3. Merged Blackbox into StaticAnalysis branch

- The merge showed "Already up to date" since both branches had the same changes

```
● $ git merge Blackbox
Already up to date.
```

```
$ git add gradlew gradlew.bat gradle/wrapper
warning: in the working copy of 'gradle/wrapper/gradle-wrapper.properties', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'gradlew', LF will be replaced by CRLF the next time Git touches it
```

```
● $ git commit -m "Added Gradle wrapper"
[StaticAnalysis 83238b8] Added Gradle wrapper
 3 files changed, 34 insertions(+), 24 deletions(-)
```

### Which branches fail on GitHub Actions? Explain why!

#### 1. Whitebox Branch:

- This branch will fail because it has the BlackBoxGiven.java file deleted
- The build will fail during compilation because the test class is missing

#### 4. Review Branch:

- This branch might fail if it doesn't have the Gradle wrapper files
- It might also fail if it has the original BlackBoxGiven.java with references to Game0-Game4 classes

### Task 1: Style checking using Checkstyle

#### Checkstyle Results

##### Summary

Total files checked	Total violations	Files with violations
2	12	2

##### Violations

File	Total violations
C:\Users\ahdma\OneDrive\Documents\2nd CSE\Eyad\user316-summer25C-eghanem-main\ser316-summer25C-eghanem-main\src\main\java\Game.java	10
C:\Users\ahdma\OneDrive\Documents\2nd CSE\Eyad\user316-summer25C-eghanem-main\ser316-summer25C-eghanem-main\src\main\java\Main.java	2

  

Severity	Description	Line Number
warning	Line is longer than 100 characters (found 152).	16
warning	First sentence of Javadoc is missing an ending period.	40
warning	'else' construct must use '{}'s.	107
warning	First sentence of Javadoc is missing an ending period.	111
warning	First sentence of Javadoc is missing an ending period.	147
warning	First sentence of Javadoc is missing an ending period.	156
warning	Line is longer than 100 characters (found 115).	171
warning	Line is longer than 100 characters (found 120).	178
warning	'{' at column 9 should be on the same line as the next part of a multi-block statement (one that directly contains multiple blocks: if/else-if/else, do/while or try/catch/finally).	237
warning	First sentence of Javadoc is missing an ending period.	280

[Back to top](#)

## **Task 2: Static Analysis using Spotbug**

### **SpotBugs Report**

#### **Project Information**

Project: ser316-summer25C-eghanem-main (spotbugsMain)

SpotBugs version: 4.8.3

Code analyzed:

- C:\Users\ahdma\OneDrive\Documents\2nd CSE\Eyad\ser316-summer25C-eghanem-main\ser316-summer25C-eghanem-main\build\classes\java\main\Game.class
- C:\Users\ahdma\OneDrive\Documents\2nd CSE\Eyad\ser316-summer25C-eghanem-main\ser316-summer25C-eghanem-main\build\classes\java\main>Main.class

#### **Metrics**

146 lines of code analyzed, in 2 classes, in 1 packages.

Metric	Total	Density*
High Priority Warnings	1	6.85
Medium Priority Warnings	3	20.55
<b>Total Warnings</b>	<b>4</b>	<b>27.40</b>

(\* Defects per Thousand lines of non-commenting source statements)

After solving the errors:

### **SpotBugs Report**

#### **Project Information**

Project: ser316-summer25C-eghanem-main (spotbugsMain)

SpotBugs version: 4.8.3

Code analyzed:

- C:\Users\ahdma\OneDrive\Documents\2nd CSE\Eyad\ser316-summer25C-eghanem-main\ser316-summer25C-eghanem-main\build\classes\java\main\Game.class
- C:\Users\ahdma\OneDrive\Documents\2nd CSE\Eyad\ser316-summer25C-eghanem-main\ser316-summer25C-eghanem-main\build\classes\java\main>Main.class

#### **Metrics**

143 lines of code analyzed, in 2 classes, in 1 packages.

Metric	Total	Density*
High Priority Warnings		0.00
Medium Priority Warnings		0.00
<b>Total Warnings</b>	<b>0</b>	<b>0.00</b>

(\* Defects per Thousand lines of non-commenting source statements)

```
$ ./gradlew spotbugsMain

> Task :spotbugsMain
Scanning archives (5 / 5)
2 analysis passes to perform
Pass 1: Analyzing classes (26 / 26) - 100% complete
Pass 2: Analyzing classes (2 / 2) - 100% complete
```

#### **Task 4: Putting it together**

##### **a) What do you get now for these branches?**

The **current branch** (after corrections) shows the following:

- **SpotBugs:** 0 warnings (both high and medium priority).
- **Checkstyle:** 12 warnings remain (mostly minor style issues, like long lines or missing Javadoc periods).

##### **b) Number of Checkstyle and SpotBugs Violations/Bugs (Current vs. Initial StaticAnalysis Branch)**

Tool	StaticAnalysis (Initial)	Current Branch
SpotBugs	4 total (1 High, 3 Medium)	0
Checkstyle	12 (10 in Game.java, 2 in Main.java)	12 (same warnings remain)

##### **c) Comparison to StaticAnalysis Branch After Corrections**

Tool	StaticAnalysis (After Fixes)	Current Branch
SpotBugs	0	0
Checkstyle	12	12

The **SpotBugs issues were resolved**, indicating functional improvements. However, the **Checkstyle violations remained unchanged**, likely because they were related to style/documentation and not fixed.

d) Table Summary of Violations Across Branches

Branch	SpotBugs (High/Medium)	SpotBugs Total	Checkstyle Total	Notes
<b>StaticAnalysis (Before Fixes)</b>	1 / 3	4	12	Functional + Style issues present
<b>StaticAnalysis (After Fixes)</b>	0 / 0	0	12	Functional bugs fixed only
<b>Current Branch (Post-Fix)</b>	0 / 0	0	12	Same as after fixes
<b>Blackbox</b>	Not shown	Not shown	Not shown	Not evaluated here
<b>Review</b>	Not shown	Not shown	Not shown	Not evaluated here

e) Did it get better or worse? Can you explain why?

- **Better** in terms of **SpotBugs** — all critical and medium issues have been resolved, indicating improvement in the logic and reliability of the code.
- **Same** in terms of **Checkstyle** — cosmetic and documentation warnings were not addressed, so this area hasn't improved.

**Why?**

The improvements were likely targeted at **functional bugs (SpotBugs)** rather than **style (Checkstyle)**. The development effort was focused on correctness and performance rather than formatting and documentation compliance.