

Static Analysis Report

Findbugs

Configuration

Findbugs was executed with the following configuration.

Analysis effort: maximal

Minimum rank to report: 20

Minimum confidence to report: Minimum

Reported bug categories:

- Bad Practice
- Malicious Code Vulnerability
- Correctness
- Performance
- Security
- Dodgy Code
- Experimental
- Multithreaded Correctness
- Internationalization

Summary

The following classes of warnings were found and remain unresolved.

- High Confidence
 - Dead store to local variable (1)
 - Reliance on default encoding (3)
- Normal Confidence
 - Field isn't final and can't be protected from malicious code (36)
 - Field should be package protected (16)
 - Return value of method without side effect is ignored (3)
 - Write to static field from instance method (1)

The reason that these warnings were found – and why they remain unsolved - will be outlined in the following section.

Dead store to local variable

This warning is found in the tests, and results in an exception test that is expected to throw an exception when the constructor is called. There is no need to take further action in this test, so the variable is not used.

Reliance on default encoding

The console outlet of the logger depends on the System.out stream. This is the only way to print to the console and these warnings can therefore be discarded.

Field isn't final and can't be protected from malicious code

The Resources class aggregates all the images from the original Bubble Trouble game. This class causes a lot of warnings. This class should be refactored into a testable utility class. While these fields

Group 4

Bubble Trouble

can easily be made final (the images do not ever change), the current structure of this class causes other errors when the fields are made final.

Field should be package protected

These warnings are like the previous, caused by the Resources class. Because this class causes so many warnings, it is evident that it requires a major overhaul and trying to solve these warnings would be harder than actually refactoring it completely.

Return value of method without side effect is ignored

A few tests assert the return value of a getter. This value is not stored outside of the assertion, and this warning is caused. In normal code, this would be relevant, but in the test it is not.

Write to static field from instance method

This is a style warning and does not have any impact on performance or readability of the source code.

Conclusion

From the Findbugs report, it can be concluded that the Resources class requires some major rework and should be prioritized in the next sprint.

PMD

Configuration

PMD was executed with the following configuration.

Rulesets:

- Java/basic
- Java/naming
- Java/design

Summary

The following classes of warnings were found and remain unsolved.

- GodClass
- LongVariable
- ShortVariable
- ShortClassName
- UseUtilityClass
- AvoidDeeplyNestedIfStmts
- SimpleDateFormatNeedsLocale
- Confusing Ternary

These warnings will be discussed in the following sections.

GodClass

PMD outlines very well that the Player class has too many responsibilities. This is very clear from the size of the class, and it would be advisable to refactor this as soon as possible into more maintainable and better structured code.

LongVariable

This warning is triggered when the name of a variable is too long. This is a warning that we selectively decided to ignore. The remaining warnings are on variable names that need the length to be understandable.

ShortVariable

Like LongVariable, ShortVariable is triggered when a variable name is too short. This can sometimes be perfectly fine however. A good example is the name 'id'. This is clear and concise.

ShortClassName

PMD decides that class names 'Game' and 'Wall' are too short. We consider these names perfect.

UseUtilityClass

This warning is triggered on the resource class. PMD suggests that this class should be made into a utility class, such that the methods are final and safe against malicious use.

While this is correct, because Resources must be refactored completely, this warning is left unresolved.

AvoidDeeplyNestedIfStmts

There are a few nested if statements. This is not very readable, but sometimes one has no choice.

Group 4

Bubble Trouble

SimpleDateFormatNeedsLocale

The date object should have a locale so that it continues to work outside of the current locale. This is a low priority warning and was left unresolved because the protection it offers is not very relevant.

Confusing Ternary

This warning was triggered by an if-else structure, but this structure could not be simplified.

Conclusion

The PMD report shows very well that the Player class is too responsible. This class should therefore be refactored.

Checkstyle

Configuration

The Checkstyle file used in this project is the Google_checks checkstyle adapted to our needs. The following adaptations have been made to the checkstyle.

- Increase all indentation from 2 to 4 whitespaces
- Allow uppercase variable names on final fields.
- Enforce no missing tags in the JavaDoc (@params, @throws, @return)
- Change the enforced order of input types to static - standard java – third party
- Suppress checkstyle on .png files (this caused tab character warnings)
- Increase the max amount of capitalized characters in variable names from 1 to 3 (SEM is a valid abbreviation).

Summary

There were no warnings left unresolved in the project.