POKERGAME

Branch : main

VERSION 1.0

Code analysis

By: Administrator

2024-12-18

CONTENT

INTRODUCTION

This document contains results of the code analysis of PokerGame.

CONFIGURATION

- Quality Profiles
  - Names: Sonar way [Java]; Sonar way [XML];
  - Files: 1db8cbf0-fff4-4cc8-b8b2-22ba279701c8.json; f231f744-67af-4291-971c-0219d6a12844.json;
- Quality Gate
  - Name: Sonar way
  - File: Sonar way.xml

SYNTHESIS

ANALYSIS STATUS

Reliability	Security	Security Review	Maintainability
A	A	A	A

QUALITY GATE STATUS

Quality Gate Status	Passed

Metric	Value
Duplicated Lines (%) on New Code	OK
New Issues	OK

METRICS

Coverage	Duplication	Comment density	Median number of lines of code per file	Adherence to coding standard
77.1 %	3.1 %	1.7 %	81.5	99.8 %

TESTS

Total	Success Rate	Skipped	Errors	Failures
34	100.0 %	0	0	0

DETAILED TECHNICAL DEBT			
Reliability	**Security**	**Maintainability**	**Total**
-	-	0d 7h 16min	**0d 7h 16min**

METRICS RANGE

	Cyclomatic Complexity	Cognitive Complexity	Lines of code per file	Comment density (%)	Coverage	Duplication (%)
Min	0.0	0.0	6.0	0.0	28.4	0.0
Max	358.0	436.0	1203.0	25.0	100.0	16.9

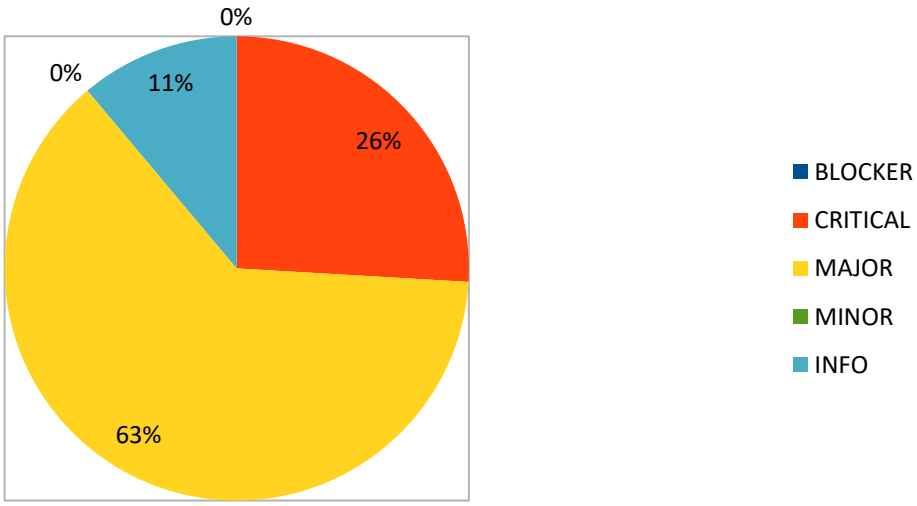
VOLUME

Language	Number
Java	1566
XML	444
Total	2010

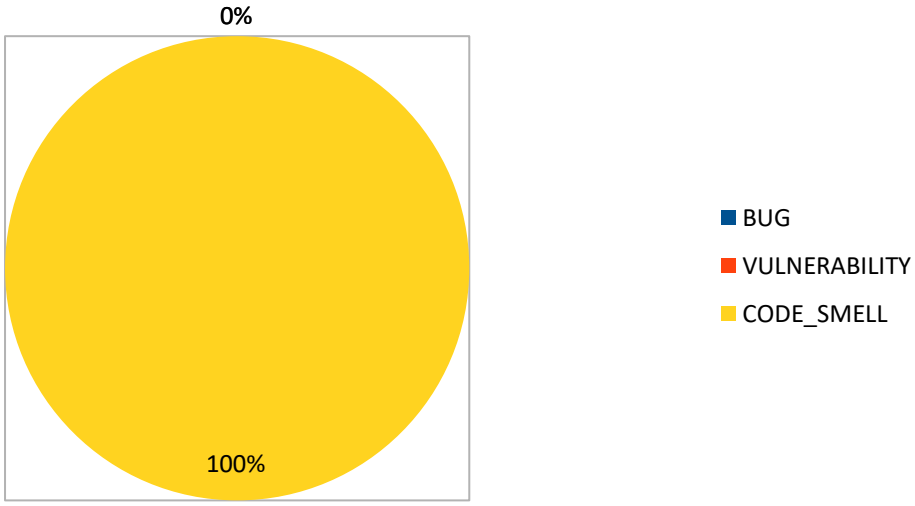
ISSUES

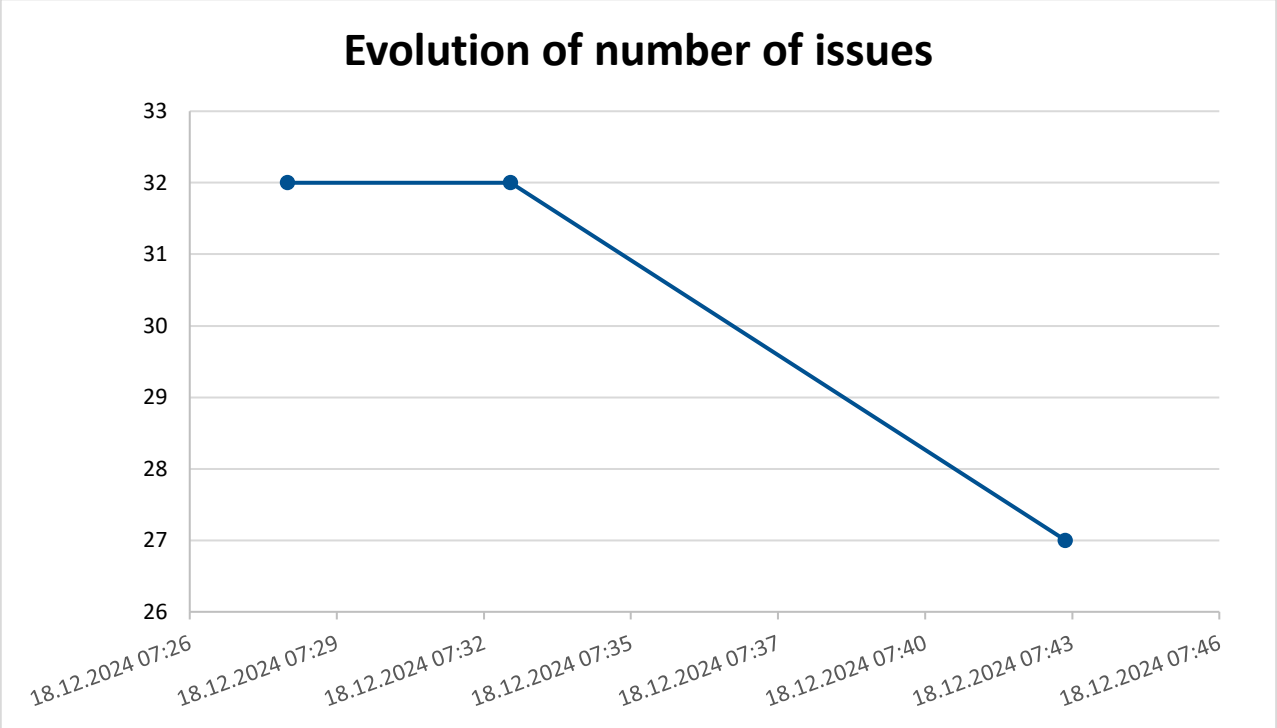
CHARTS

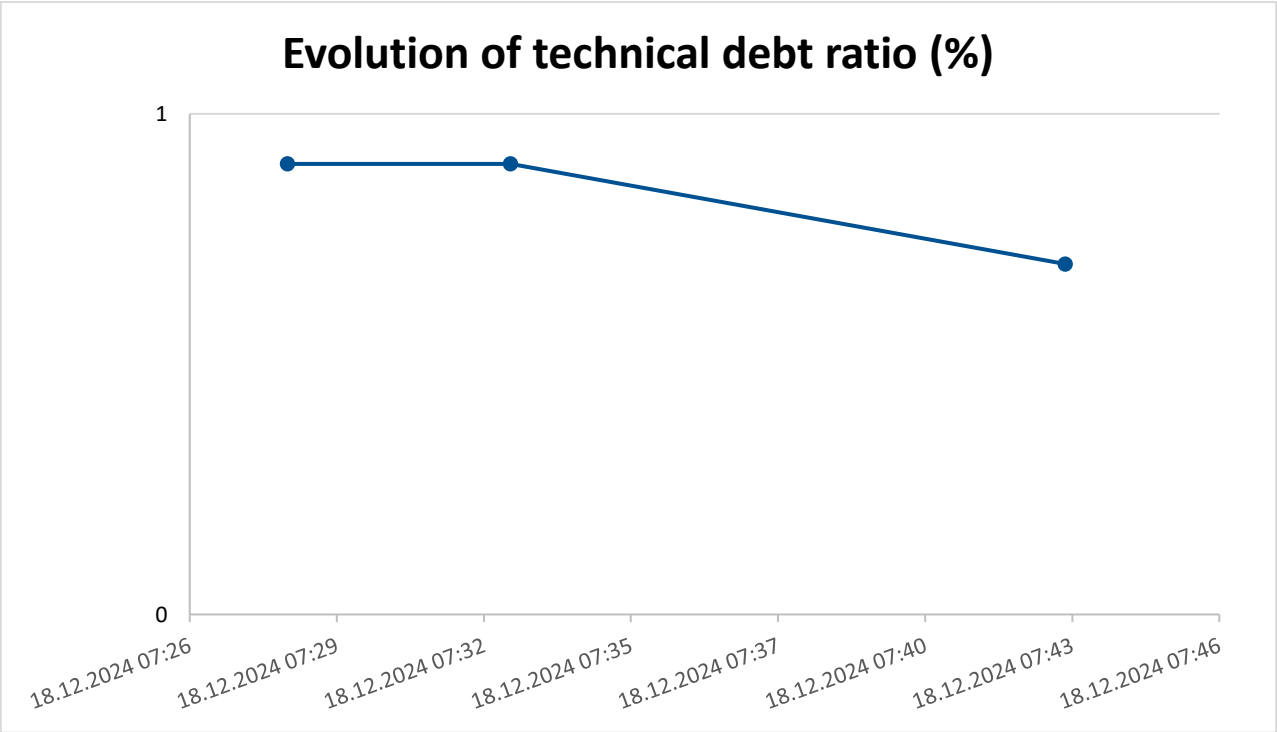
Number of issues by severity



Number of issues by type




Evolution of number of issues
33
32
31
30
29
28
27
26
18.12.2024 07:26
18.12.2024 07:29
18.12.2024 07:32
18.12.2024 07:35
18.12.2024 07:37
18.12.2024 07:40
18.12.2024 07:43
18.12.2024 07:46


Evolution of technical debt ratio (%)
1
0
18.12.2024 07:26
18.12.2024 07:29
18.12.2024 07:32
18.12.2024 07:35
18.12.2024 07:37
18.12.2024 07:40
18.12.2024 07:43
18.12.2024 07:46

ISSUES COUNT BY SEVERITY AND TYPE

Type / Severity	INFO	MINOR	MAJOR	CRITICAL	BLOCKER
BUG	0	0	0	0	0
VULNERABILITY	0	0	0	0	0
CODE_SMELL	3	0	17	7	0

ISSUES LIST

Name	Description	Type	Severity	Number
Cognitive Complexity of methods should not be too high		CODE_SMELL	CRITICAL	7
Methods should not perform too many tasks (aka Brain method)		CODE_SMELL	INFO	3
Standard outputs should not be used directly to log anything		CODE_SMELL	MAJOR	14
"for" loop stop conditions should be invariant		CODE_SMELL	MAJOR	1
"entrySet()" should be iterated when both the key and value are needed		CODE_SMELL	MAJOR	1
Similar tests should be grouped in a single Parameterized test		CODE_SMELL	MAJOR	1

SECURITY HOTSPOTS

SECURITY HOTSPOTS COUNT BY CATEGORY AND PRIORITY

Category / Priority	LOW	MEDIUM	HIGH
LDAP Injection	0	0	0
Object Injection	0	0	0
Server-Side Request Forgery (SSRF)	0	0	0
XML External Entity (XXE)	0	0	0
Insecure Configuration	0	0	0
XPath Injection	0	0	0
Authentication	0	0	0
Weak Cryptography	0	0	0
Denial of Service (DoS)	0	0	0
Log Injection	0	0	0
Cross-Site Request Forgery (CSRF)	0	0	0
Open Redirect	0	0	0
Permission	0	0	0
SQL Injection	0	0	0
Encryption of Sensitive Data	0	0	0
Traceability	0	0	0
Buffer Overflow	0	0	0
File Manipulation	0	0	0
Code Injection (RCE)	0	0	0

Cross-Site Scripting (XSS)	0	0	0
Command Injection	0	0	0
Path Traversal Injection	0	0	0
HTTP Response Splitting	0	0	0
Others	0	0	0

SECURITY HOTSPOTS LIST