



**INSPECTOR
LOVELY**

SMART CONTRACT

SECURITY AUDIT

MAD VIKING GAMES



inspector.lovely.finance





TABLE OF CONTENTS

Table of Contents	2
Disclaimer	3
Audit Scope	4
Proposed Smart Contract Features	5
Audit Summary	6
Key Technical Metrics	7
Code Quality	8
Documentation	8
Use of Dependencies	8
AS-IS Overview	9
Project Website Performance Audit	11
Level of Criticality	13
Audit Findings	14
Centralization	15
Conclusion	16
• Logic Diagram	17
• Security Assessment Report	19
• Solidity Static Analysis	20
• Compliance Analysis	22
Software Analysis Result	22
INSPECTOR Lovely Info	23





**INSPECTOR
LOVELY**

DISCLAIMER

This is a comprehensive report based on our automated and manual examination of cybersecurity vulnerabilities and framework flaws of the project's smart contract. Reading the full analysis report is essential to build your understanding of project's security level. It is crucial to take note, though we have done our best to perform this analysis and report, that you should not rely on the our research and cannot claim what it states or how we created it. Before making any judgments, you have to conduct your own independent research. We will discuss this in more depth in the following disclaimer - please read it fully. **DISCLAIMER:** You agree to the terms of this disclaimer by reading this report or any portion thereof. Please stop reading this report and remove and delete any copies of this report that you download and/or print if you do not agree to these conditions. Scan and verify report's presence in the GitHub repository by a qr-code on the title page. This report is for non-reliability information only and does not represent investment advice. No one shall be entitled to depend on the report or its contents, and Inspector Lovely and its affiliates shall not be held responsible to you or anyone else, nor shall Inspector Lovely provide any guarantee or representation to any person with regard to the accuracy or integrity of the report. Without any terms, warranties or other conditions other than as set forth in that exclusion and Inspector Lovely excludes hereby all representations, warrants, conditions and other terms (including, without limitation, guarantees implied by the law of satisfactory quality, fitness for purposes and the use of reasonable care and skills). The report is provided as "as is" and does not contain any terms and conditions. Except as legally banned, Inspector Lovely disclaims all responsibility and responsibilities and no claim against Inspector Lovely is made to any amount or type of loss or damages (without limitation, direct, indirect, special, punitive, consequential or pure economic loses or losses) that may be caused by you or any other person, or any damages or damages, including without limitations (whether innocent or negligent). Security analysis is based only on the smart contracts. No applications or operations were reviewed for security. No product code has been reviewed.



**INSPECTOR
LOVELY**



AUDIT SCOPE

Name	Code Review and Security Analysis Report for Mad Viking Games Smart Contracts
Platform	BSC / Solidity
File 1	GEMS.sol
File 1 MD5 Hash	D2F248F722DB3B87C6E0D303AFF259BB
File 2	MVG.sol
File 2 MD Hash	A8F6BD45F8B053B8A630D6725D613876
Audit Date	March 11th,2023



inspector.lovely.finance

Audited by INSPECTOR LOVELY



PROPOSED SMART CONTRACT FEATURES

Claimed Feature Detail	Our Observation
<p>File 1 GEMS.sol</p> <p>Tokenomics:</p> <ul style="list-style-type: none">• Name: GEMS• Symbol: GEMS• Decimals: 18• Total Supply: 100 Billion	Validated
<p>File 2 MVG.sol</p> <p>Tokenomics:</p> <ul style="list-style-type: none">• Name: Mad Viking Games• Symbol: MVG• Decimals: 18• Total Supply: 14 Billion	Validated

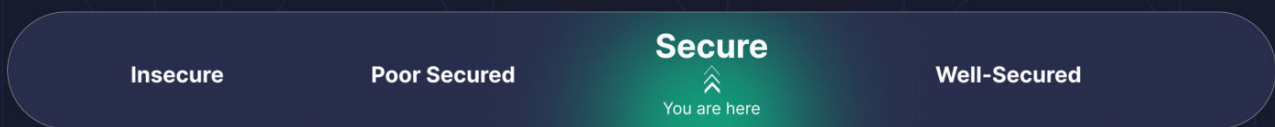




**INSPECTOR
LOVELY**

AUDIT SUMMARY

According to the standard audit assessment, Customer`s solidity smart contracts are “Secured”. Also, these contracts do contain owner control, which does not make them fully decentralized.



We used various tools like Slither, Solhint, and Remix IDE. At the same time, this finding is based on a critical analysis of the manual audit.

All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the Audit Overview section. General overview is presented in the AS-IS section and all identified issues can be found in the Audit overview section.

We found 0 critical, 0 high, 0 medium, and 0 low, and some very low-level issues.

Investors Advice: A technical audit of the smart contract does not guarantee the ethical nature of the project. Any owner-controlled functions should be executed by the owner with responsibility. All investors/users are advised to do their due diligence before investing in the project.



inspector.lovely.finance

Audited by INSPECTOR LOVELY



KEY TECHNICAL METRICS

MAIN CATEGORY	SUBCATEGORY	RESULT
Contract Programming	Solidity version is not specified	Passed
	Solidity version is too old	Passed
	Integer overflow/underflow	Passed
	Function input parameters lack check	Passed
	Function input parameters check bypass	Passed
	Function access control lacks management	Passed
	Critical operation lacks event log	Passed
	Human/contract checks bypass	Passed
	Random number generation/use vulnerability	N/A
	Fallback function misuse	Passed
	Race condition	Passed
	Logical vulnerability	Passed
	Features claimed	Passed
Other programming issues	Passed	
Code Specification	Function visibility not explicitly declared	Passed
	Var. storage location not explicitly declared	Passed
	Use keywords/functions to be deprecated	Passed
	Unused code	Passed
Gas Optimization	"Out of Gas" Issue	Passed
	High consumption 'for/while' loop	Passed
	High consumption 'storage' storage	Passed
	Assert() misuse	Passed
Business Risk	The maximum limit for mintage is not set	Passed
	"Short Address" Attack	Passed
	"Double Spend" Attack	Passed

Overall Audit Result: PASSED



CODE QUALITY

This audit scope has 2 smart contract files. Smart contracts contain Libraries, Smart contracts, inherits, and Interfaces. This is a compact and well-written smart contract.

The libraries in the Mad Viking Games Protocol are part of its logical algorithm. A library is a different type of smart contract that contains reusable code. Once deployed on the blockchain (only once), it is assigned a specific address and its properties/methods can be reused many times by other contracts in the Mad Viking Games Protocol.

The Mad Viking Games team has not provided unit test scripts, which would have helped to determine the integrity of the code in an automated way.

Code parts are well commented on smart contracts.

DOCUMENTATION

We were given a Mad Viking Games Protocol smart contract code in the form of a file. The hash of that code is mentioned above in the table.

As mentioned above, code parts are well-commented. So it is easy to quickly understand the programming flow as well as complex code logic. Comments are very helpful in understanding the overall architecture of the protocol.

Another source of information was its official website: <https://madvikingstudios.com> which provided rich information about the project architecture and tokenomics.

USE OF DEPENDENCIES

As per our observation, the libraries are used in this smart contracts infrastructure that are based on well-known industry-standard open-source projects.

Apart from libraries, its functions are used in external smart contract calls.





AS-IS OVERVIEW

GEMS.sol

Functions

SL.	FUNCTIONS	TYPE	OBSERVATION	CONCLUSION
1	constructor	write	Passed	No Issue
2	onlyOwner	modifier	Passed	No Issue
3	owner	read	Passed	No Issue
4	_checkOwner	internal	Passed	No Issue
5	renounceOwnership	write	access only Owner	No Issue
6	transferOwnership	write	access only Owner	No Issue
7	_transferOwnership	internal	Passed	No Issue
8	name	read	Passed	No Issue
9	symbol	read	Passed	No Issue
10	decimals	read	Passed	No Issue
11	totalSupply	read	Passed	No Issue
12	balanceOf	read	Passed	No Issue
13	transfer	write	Passed	No Issue
14	allowance	read	Passed	No Issue
15	approve	write	Passed	No Issue
16	transferFrom	write	Passed	No Issue
17	increaseAllowance	write	Passed	No Issue
18	decreaseAllowance	write	Passed	No Issue
19	_transfer	internal	Passed	No Issue
20	_mint	internal	Passed	No Issue
21	_burn	internal	Passed	No Issue
22	_approve	internal	Passed	No Issue
23	_spendAllowance	internal	Passed	No Issue
24	_beforeTokenTransfer	internal	Passed	No Issue
25	_afterTokenTransfer	internal	Passed	No Issue





MVG.sol

Functions

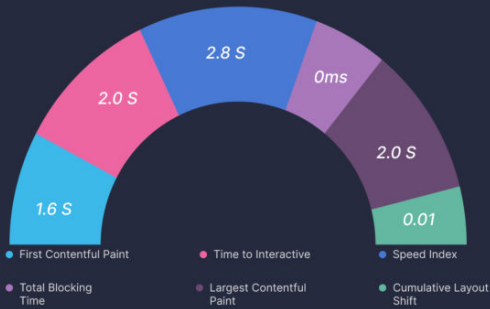
SL.	FUNCTIONS	TYPE	OBSERVATION	CONCLUSION
1	constructor	write	Passed	No Issue
2	onlyOwner	modifier	Passed	No Issue
3	owner	read	Passed	No Issue
4	_checkOwner	internal	Passed	No Issue
5	renounceOwnership	write	access only Owner	No Issue
6	transferOwnership	write	access only Owner	No Issue
7	_transferOwnership	internal	Passed	No Issue
8	name	read	Passed	No Issue
9	symbol	read	Passed	No Issue
10	decimals	read	Passed	No Issue
11	totalSupply	read	Passed	No Issue
12	balanceOf	read	Passed	No Issue
13	transfer	write	Passed	No Issue
14	allowance	read	Passed	No Issue
15	approve	write	Passed	No Issue
16	transferFrom	write	Passed	No Issue
17	increaseAllowance	write	Passed	No Issue
18	decreaseAllowance	write	Passed	No Issue
19	_transfer	internal	Passed	No Issue
20	_mint	internal	Passed	No Issue
21	_burn	internal	Passed	No Issue
22	_approve	internal	Passed	No Issue
23	_spendAllowance	internal	Passed	No Issue
24	_beforeTokenTransfer	internal	Passed	No Issue
25	_afterTokenTransfer	internal	Passed	No Issue





PROJECT WEBSITE PERFORMANCE AUDIT

Performance Metrics



Browser Timings

Redirect Duration	0ms	Connection Duration	175ms	Backend Duration	677ms
Time to First Byte	852s	First Paint	1.6s	DOM Interactive Time	2.0s
DOM Content Loaded	2.0s	Onload Time	2.8s	Fully Loaded Time	2.9s

Grade

C

Performance	79%
Structure	68%

Top Issues

IMPECT

AUDIT

High

Avoid enormous network payloads (LCP)

Total size was 21 MB

URL

- https://madvikingstudios.com/wp-content/uploads/2022/09/Background_video_northern_lights.mp4
- https://madvikingstudios.com/wp-content/uploads/2022/09/Sword_and_helmet_wide_screen_min-scaled.webp
- https://www.googletagmanager.com/gtag/js?id=G-M471JZ9S5J
- https://www.googletagmanager.com/gtag/js?id=G-M471JZ9S5J&l=dataLayer&cx=c
- https://madvikingstudios.com/wp-content/plugins/elementor/assets/lib/font-awesome/webfonts/fa-solid-900.woff2
- https://madvikingstudios.com/wp-content/plugins/elementor/assets/lib/font-awesome/webfonts/fa-brands-400.woff2
- https://www.googletagmanager.com/gtag/js?id=UA-240307462-1
- https://www.googletagmanager.com/gtag/js?id=UA-240307462-1&l=dataLayer&cx=c
- https://connect.facebook.net/en_US/fbevents.js
- https://www.googleoptimize.com/optimize.js?id=OPT-KCRJS3N

SIZE

- 19.6MB
- 179KB
- 85.2KB
- 84.1KB
- 76.6KB
- 75.2KB
- 67.9KB
- 67.3KB
- 53.3KB
- 49.9KB

High

Reduce initial server response time (RCP) (LCP)

Root document took 677ms

URL

- https://madvikingstudios.com/

SIZE

- 677ms





Top Issues

IMPACT

AUDIT

High

Eliminate render-blocking resources (RCP) (LCP)

Potential savings of 1.9s

URL	SIZE	DOWNLOAD TIME
https://madvikingstudios.com/wp-includes/css/dist/block-library/style.min.css?ver=6.0.6	11.7KB	669ms
https://madvikingstudios.com/wp-content/themes/hello-elementor/style.min.css?ver=2.6.1	2.21KB	669ms
https://madvikingstudios.com/wp-content/themes/hello-elementor/theme.min.css?ver=2.6.1	2.92KB	669ms
https://madvikingstudios.com/wp-content/uploads/elementor/css/custom-front-end-lite.min.css?ver=1680292190	10.8KB	1.1s
https://madvikingstudios.com/wp-content/uploads/elementor/css/post-4545.css?ver=1680292190	903B	669ms
https://madvikingstudios.com/wp-content/plugins/elementor/assets/lib/eicons/css/elementor-icons.min.css?ver=5.18.0	4.20KB	669ms
https://madvikingstudios.com/wp-content/plugins/elementor/assets/lib/swiper/css/swiper.min.css?ver=5.3.6	2.67KB	669ms
https://madvikingstudios.com/wp-content/uploads/elementor/css/custom-pro-front-end-lite.min.css?ver=1680292190	1.92KB	669ms
https://madvikingstudios.com/wp-content/plugins/elementor/assets/lib/font-awesome/css/all.min.css?ver=3.12.0	12.9KB	1.3s
https://madvikingstudios.com/wp-content/plugins/elementor/assets/lib/font-awesome/css/v4-shims.min.css?ver=3.12.0	4.44KB	892ms
https://madvikingstudios.com/wp-content/uploads/elementor/css/global.css?ver=1680292193	4.70KB	892ms
https://madvikingstudios.com/wp-content/uploads/elementor/css/post-4584.css?ver=1681133206	2.66KB	669ms
https://madvikingstudios.com/wp-content/uploads/elementor/css/post-4547.css?ver=1680292193	1.75KB	669ms
https://madvikingstudios.com/wp-content/plugins/happy-elementor-addons/assets/fonts/style.min.css?ver=3.8.3	5.50KB	892ms
https://madvikingstudios.com/wp-content/plugins/elementor/assets/lib/font-awesome/css/font-awesome.min.css?ver=4.7.0	7.20KB	892ms
https://madvikingstudios.com/wp-content/uploads/elementor/css/post-4550.css?ver=1681481765	2.40KB	669ms
https://madvikingstudios.com/wp-content/uploads/elementor/css/post-4557.css?ver=1680292202	728B	669ms
https://madvikingstudios.com/wp-content/uploads/happyaddons/css/ha-4584.css?ver=3.8.3.1681133206	1.42KB	669ms
https://fonts.googleapis.com/css?family=Roboto%3A100%2C100italic%2C200%2C200italic%2C300%2C300italic%2C400%2C400italic%2C500%2C500italic%2C600%2C600italic%2C700%2C700italic%2C800%2C800italic%2C900%2C900italic%7CRoboto+Slab%3A100%2C100italic%2C200%2C200italic%2C300%2C300italic%2C400%2C400italic%2C500%2C500italic%2C600%2C600italic%2C700%2C700italic%2C800%2C800italic%2C900%2C900italic%7C Oswald%3A100%2C100italic%2C200%2C200italic%2C300%2C300italic%2C400%2C400italic%2C500%2C500italic%2C600%2C600italic%2C700%2C700italic%2C800%2C800italic%2C900%2C900italic%7CPlay%3A100%2C100italic%2C200%2C200italic%2C300%2C300italic%2C400%2C400italic%2C500%2C500italic%2C600%2C600italic%2C700%2C700italic%2C800%2C800italic%2C900%2C900italic%7CMontserrat%3A100%2C100italic%2C200%2C200italic%2C300%2C300italic%2C400%2C400italic%2C500%2C500italic%2C600%2C600italic%2C700%2C700italic%2C800%2C800italic%2C900%2C900italic&display=auto&ver=6.0.6	2.99KB	921ms
https://madvikingstudios.com/wp-content/plugins/elementor/assets/lib/font-awesome/css/font-awesome.min.css?ver=5.15	12.6KB	1.1s
https://madvikingstudios.com/wp-content/plugins/elementor/assets/lib/font-awesome/css/solid.min.css?ver=5.15.3	628B	669ms
https://madvikingstudios.com/wp-content/plugins/elementor/assets/lib/font-awesome/css/brands.min.css?ver=5.15.3	627B	669ms
https://madvikingstudios.com/wp-content/uploads/elementor/css/custom-pro-widget-nav-menu.min.css?ver=1680292194	3.75KB	669ms
https://madvikingstudios.com/wp-content/uploads/elementor/css/custom-widget-icon-list.min.css?ver=1680292195	1.24KB	669ms
https://madvikingstudios.com/wp-includes/js/jquery/jquery.min.js?ver=3.6.0	30.5KB	1.8s
https://madvikingstudios.com/wp-includes/js/jquery/jquery-migrate.min.js?ver=3.3.2	4.40KB	892ms
https://madvikingstudios.com/wp-content/plugins/elementor-pro/assets/js/page-transitions.min.js?ver=3.7.7	22.4KB	1.6s
https://madvikingstudios.com/wp-content/plugins/elementor/assets/lib/font-awesome/js/v4-shims.min.js?ver=3.12.0	4.43KB	892ms
https://madvikingstudios.com/wp-content/plugins/pixelyoursite/dist/scripts/jquery.bind-first-0.2.3.min.js?ver=6.0.6	1.00KB	669ms
https://madvikingstudios.com/wp-content/plugins/pixelyoursite/dist/scripts/js.cookie-2.1.3.min.js?ver=2.1.3	1.18KB	669ms
https://madvikingstudios.com/wp-content/plugins/pixelyoursite/dist/scripts/public.js?ver=9.3.5	17.0KB	892ms





Level of Criticality

RISK LEVEL	DESCRIPTION
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to token loss etc.
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g. public access to crucial
Med	Medium-level vulnerabilities are important to fix; however, they can't lead to tokens loss
Low	Low-level vulnerabilities are mostly related to outdated, unused etc. code snippets, that can't have a significant impact on execution
Lowest / Code Style / Best Practice	Lowest-level vulnerabilities, code style violations, and info statements can't affect smart contract execution and can be ignored.





AUDIT FINDINGS

Critical Severity

No Critical-severity vulnerabilities were found.

High Severity

No High-severity vulnerabilities were found.

Medium

No Medium-severity vulnerabilities were found.

Low

No low-severity vulnerabilities were found.

VERY LOW / INFORMATIONAL / BEST PRACTICES:

(1) Multiple Pragma: MVG.sol, GEMS.sol

```
5 // OpenZeppelin Contracts v4.4.1 (utils/Context.sol)
6
7 pragma solidity ^0.8.0;
8
9 > /** ...
19 > abstract contract Context { ...
27 }
28
29 // File: @openzeppelin/contracts/access/Ownable.sol
30
31
32 // OpenZeppelin Contracts (last updated v4.7.0) (access/Ownable.sol)
33
34 pragma solidity ^0.8.0;
35 > /** ...
47 > abstract contract Ownable is Context { ...
10 }
11
12 // File: @openzeppelin/contracts/token/ERC20/IERC20.sol
13 // OpenZeppelin Contracts (last updated v4.6.0) (token/ERC20/IERC20.sol)
14 pragma solidity ^0.8.0;
```

There are multiple pragma with different versions.

Resolution: We suggest keeping one pragma line at the top of the code.

(2) SPDX license identifier missing: MVG.sol, GEMS.sol

SPDX license identifier is not provided in the source file.

Resolution: We suggest adding an SPDX license identifier.



CENTRALIZATION

This smart contract has some functions that can be executed by the Admin (Owner) only. If the admin wallet's private key would be compromised, then it would create trouble. Following are Admin functions:

Ownable.sol

- `renounceOwnership`: Deleting ownership will leave the contract without an owner, removing any owner-only functionality.
- `transferOwnership`: Current owner can transfer ownership of the contract to a new account.
- `_checkOwner`: Thrown when the sender is not the owner.

To make the smart contract 100% decentralized, we suggest renouncing ownership in the smart contract once its function is completed.



CONCLUSION

We were given a contract code in the form of files. And we have used all possible tests based on given objects as files. We had observed some informational severity issues in the smart contracts. But those are not critical ones. **So, the smart contracts are ready for the mainnet deployment.**

Since possible test cases can be unlimited for such smart contracts protocol, we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover the maximum possible test cases to scan everything.

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools. Smart Contract's high-level description of functionality was presented in the As-is overview section of the report.

Audit report contains all found security vulnerabilities and other issues in the reviewed code.

Security state of the reviewed contract, based on standard audit procedure scope, is **"Secured"**.

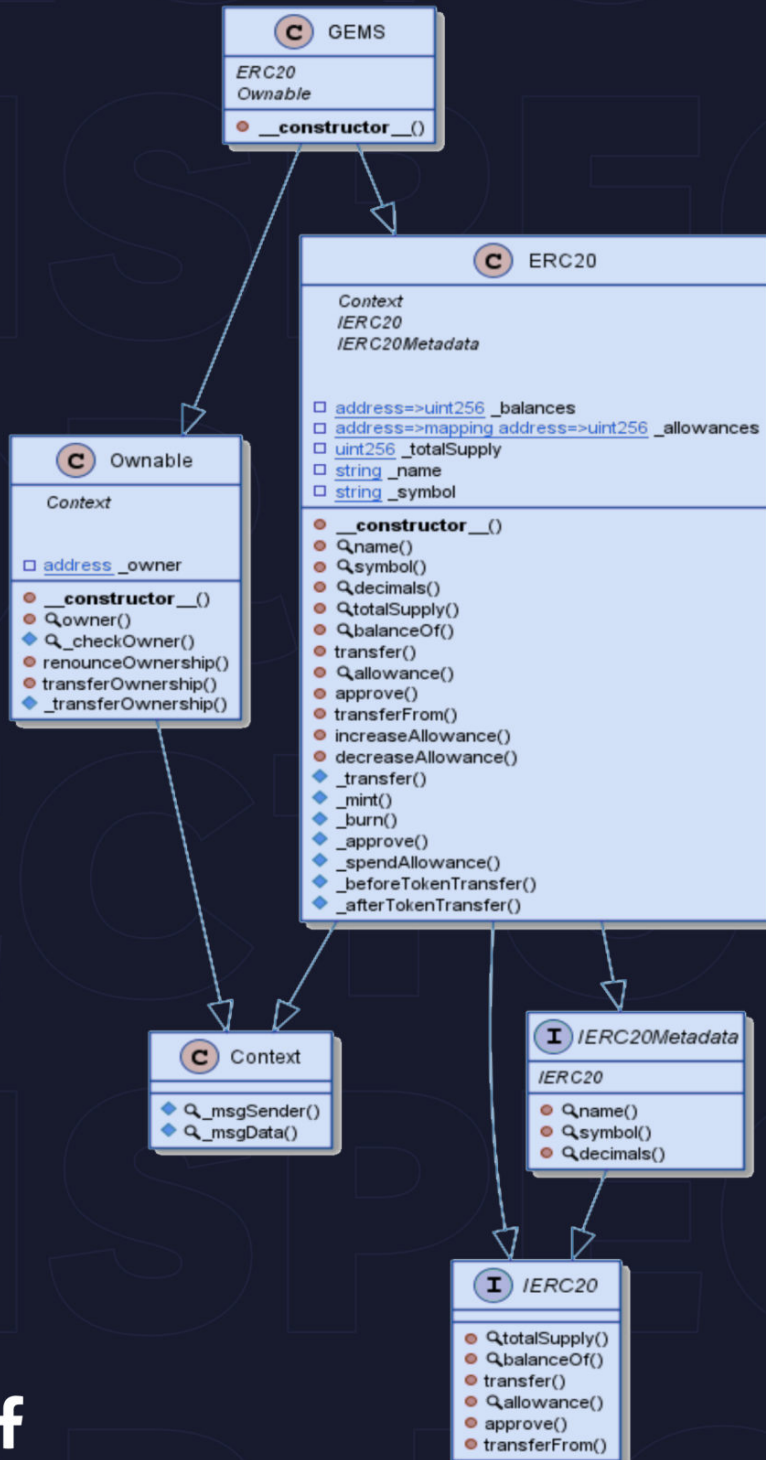




ADDENDUM

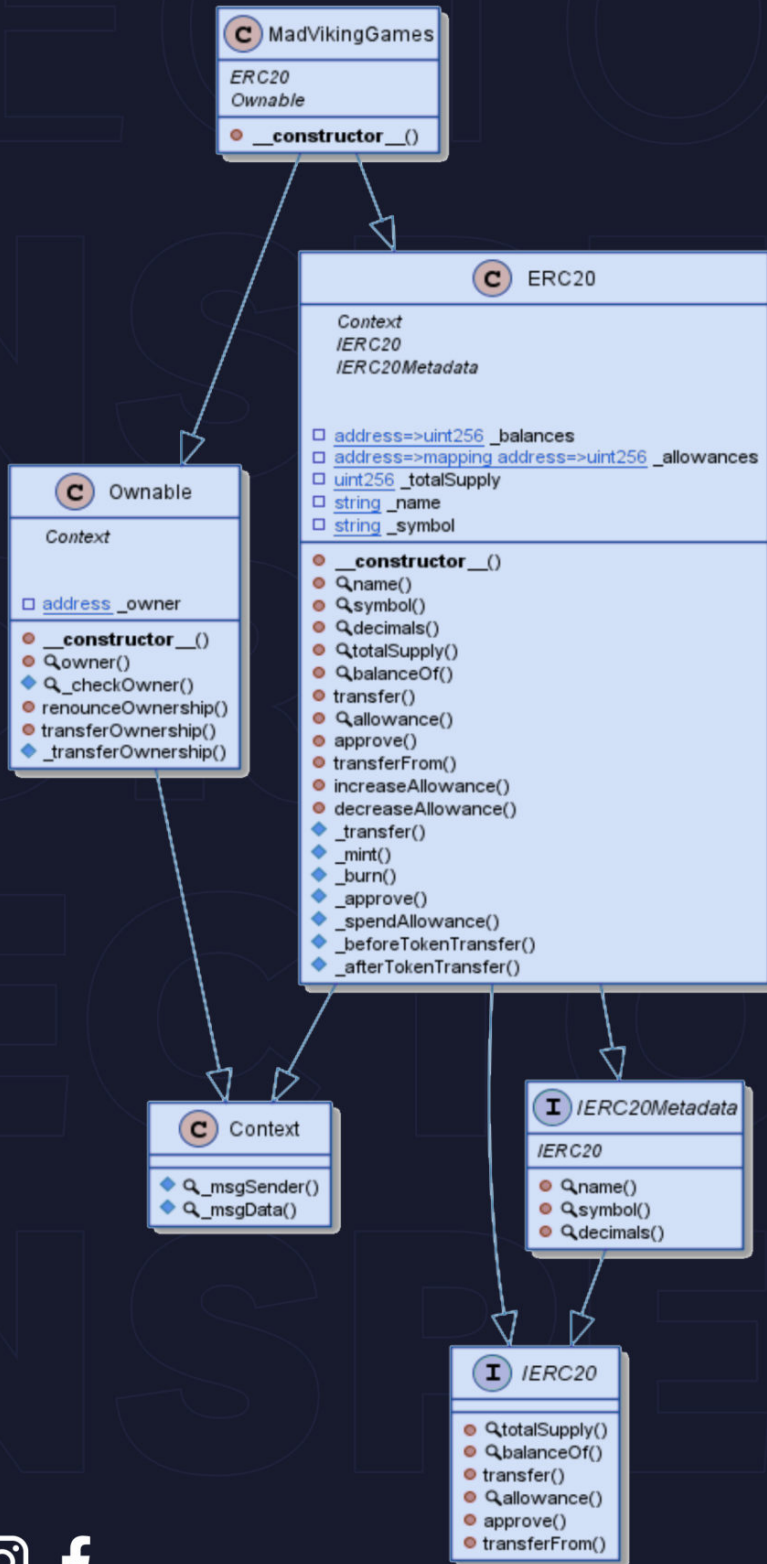
Code Flow Diagram - Mad Viking Games

GEMS Diagram





MVG Diagram





SECURITY ASSESSMENT REPORT

Slither log >> GEMS.sol

```
Context._msgData() (GEMS.sol#24-26) is never used and should be removed
ERC20._burn(address,uint256) (GEMS.sol#514-530) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (GEMS.sol#7) allows old versions
Pragma version^0.8.0 (GEMS.sol#34) allows old versions
Pragma version^0.8.0 (GEMS.sol#119) allows old versions
Pragma version^0.8.0 (GEMS.sol#204) allows old versions
Pragma version^0.8.0 (GEMS.sol#234) allows old versions
solc-0.8.0 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

GEMS.constructor() (GEMS.sol#625-627) uses literals with too many digits:
- _mint(msg.sender,100000000000 * 10 ** decimals()) (GEMS.sol#626)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
GEMS.sol analyzed (6 contracts with 84 detectors), 9 result(s) found
```

Slither log >> MVG.sol

```
Context._msgData() (MVG.sol#24-26) is never used and should be removed
ERC20._burn(address,uint256) (MVG.sol#514-530) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (MVG.sol#7) allows old versions
Pragma version^0.8.0 (MVG.sol#34) allows old versions
Pragma version^0.8.0 (MVG.sol#119) allows old versions
Pragma version^0.8.0 (MVG.sol#204) allows old versions
Pragma version^0.8.0 (MVG.sol#234) allows old versions
solc-0.8.0 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

MadVikingGames.constructor() (MVG.sol#621-623) uses literals with too many digits:
- _mint(msg.sender,140000000000 * 10 ** decimals()) (MVG.sol#622)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
MVG.sol analyzed (6 contracts with 84 detectors), 9 result(s) found
```



SOLIDITY STATIC ANALYSIS

GEMS.sol

Gas & Economy

Gas costs:

Gas requirement of function GEMS.decreaseAllowance is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 430:4:

Miscellaneous

Constant/View/Pure functions:

ERC20._afterTokenTransfer(address,address,uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 613:4:

Similar variable names:

ERC20._burn(address,uint256) : Variables have very similar names "account" and "amount". Note: Modifiers are currently not considered by this static analysis.

Pos: 529:49:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 572:12:



MVG.sol

Gas & Economy

Gas costs:

Gas requirement of function MadVikingGames.transfer is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 342:4:

Miscellaneous

Constant/View/Pure functions:

IERC20.transferFrom(address,address,uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 192:4:

Similar variable names:

ERC20._mint(address,uint256) : Variables have very similar names "account" and "amount". Note: Modifiers are currently not considered by this static analysis.

Pos: 498:43:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

[more](#)

Pos: 572:12:



COMPLIANCE ANALYSIS

GEMS.sol

```
GEMS.sol:434:18: Error: Parse error: missing ';' at '{'  
GEMS.sol:467:18: Error: Parse error: missing ';' at '{'  
GEMS.sol:494:18: Error: Parse error: missing ';' at '{'  
GEMS.sol:521:18: Error: Parse error: missing ';' at '{'  
GEMS.sol:573:22: Error: Parse error: missing ';' at '{'
```

MVG.sol

```
MVG.sol:434:18: Error: Parse error: missing ';' at '{'  
MVG.sol:467:18: Error: Parse error: missing ';' at '{'  
MVG.sol:494:18: Error: Parse error: missing ';' at '{'  
MVG.sol:521:18: Error: Parse error: missing ';' at '{'  
MVG.sol:573:22: Error: Parse error: missing ';' at '{'
```

Software analysis result:

This software reported many false positive results and some are informational issues.

So, those issues can be safely ignored.



**INSPECTOR
LOVELY**

INSPECTOR LOVELY

INFO

Website: Inspector.lovely.finance

Telegram community: t.me/inspectorlovely

Twitter: twitter.com/InspectorLovely



[inspector.lovely.finance](https://Inspector.lovely.finance)

