BlockSAFU

# ADVANCE MANUAL SMART CONTRACT AUDIT

AGRAVITY

**Project:** AgravityNFT

**Website:** https://www.agravitynft.com/

PASSED

**BlockSAFU Score:**

**85**

**Contract Address:**

**0xcb910716189ad905Deb3c7492cbD2e37EFb094B1**

Disclamer: BlockSAFU is not responsible for any financial losses.
Nothing in this contract audit is financial advice, please do your own reasearch.

# DISCLAMER

BlockSAFU has completed this report to provide a summary of the Smart Contract functions, and any security, dependency, or cybersecurity vulnerabilities. This is often a constrained report on our discoveries based on our investigation and understanding of the current programming versions as of this report’s date. To understand the full scope of our analysis, it is vital for you to at the date of this report. To understand the full scope of our analysis, you need to review the complete report. Although we have done our best in conducting our investigation and creating this report, it is vital to note that you should not depend on this report and cannot make any claim against BlockSAFU or its Subsidiaries and Team members on the premise of what has or has not been included in the report. Please remember to conduct your independent examinations before making any investment choices. We do not provide investment advice or in any way claim to determine if the project will be successful or not.

By perusing this report or any portion of it, you concur to the terms of this disclaimer. In the unlikely situation where you do not concur to the terms, you should immediately terminate reading this report, and erase and discard any duplicates of this report downloaded and/or printed by you. This report is given for data purposes as it were and on a non-reliance premise and does not constitute speculation counsel. No one should have any right to depend on the report or its substance, and BlockSAFU and its members (including holding companies, shareholders, backups, representatives, chiefs, officers, and other agents) BlockSAFU and its subsidiaries owe no obligation of care towards you or any other person, nor does BlockSAFU make any guarantee or representation to any individual on the precision or completeness of the report.

ABOUT THE AUDITOR:

BlockSAFU (BSAFU) is an Anti-Scam Token Utility that reviews Smart Contracts and Token information to Identify Rug Pull and Honey Pot scamming activity. BlockSAFUs Development Team consists of several Smart Contract creators, Auditors Developers, and Blockchain experts. BlockSAFU provides solutions, prevents, and hunts down scammers. BSAFU is a utility token with features Audit, KYC, Token Generators, and Bounty Scammers. It will enrich the crypto ecosystem.

# SMART CONTRACT REVIEW

| | |
|---|---|
| Token Name | **Agravity** |
| Token Symbol | **Agravity** |
| Total Supply | 0 **Agravity (ERC 1155)** |
| Contract Address | 0xcb910716189ad905Deb3c7492cbD2e37EFb094B1 |
| Deployer Address | 0xf330bc9EFF5a5006608784f54166c923434e3d54 |
| Owner Address | 0xf330bc9EFF5a5006608784f54166c923434e3d54 |
| Tax Fees Buy | 0% |
| Tax Fees Sell | 0% |
| Gas Used for Buy | *will be updated after the DEX listing* |
| Gas Used for Sell | *will be updated after the DEX listing* |
| Contract Created | Aug-07-2022 03:18:34 AM +UTC |
| Verified CA | Yes |
| Compiler | v0.8.9+commit.e5eed63a |
| Optimization | Yes with 200 runs |
| Sol License | None License |
| Top 5 Holders | *will be updated after the DEX listing* |
| Other | default evmVersion |

# TAX

| | | | |
|---|---|---|---|
| **BUY** | 0% | **SELL** | 0% |

## OVERVIEW

Mint Function

- Mint functions found.

Fees

- Buy 0%.
- Sell 0%.

Tx Amount

- Owner cannot set max tx amount.

Transfer Pausable

- Owner cannot pause.

Blacklist

- Owner cannot blacklist.

Ownership

- Owner cannot take back ownership.

Proxy

- This contract has no proxy.

Anti Whale

- Owner cannot limit the number of wallet holdings.

Trading Cooldown

- Owner cannot set the selling time interval.

## Team Review

The AgravityNFT team has a nice website, their website is professionally built and the Smart contract is well developed, their social media is growing with over 14,444 people in their telegram group (count in audit date).

## Official Website And Social Media

Website: https://www.agravitynft.com/

Telegram Group: https://t.me/Agravitygame

Twitter: https://twitter.com/hardcandy_lab

## MANUAL CODE REVIEW

🟢 Minor-risk

0 minor-risk code issue found

Could be fixed, and will not bring problems.

🟡 Medium-risk

0 medium-risk code issues found

Should be fixed, could bring problems.

🔴 High-Risk

0 high-risk code issues found

Must be fixed, and will bring problem.

🔴 Critical-Risk

0 critical-risk code issues found

Must be fixed, and will bring problem.

# EXTRA NOTES SMART CONTRACT

## 1. ERC1155

```
interface IERC1155 is IERC165 {
    /**
     * @dev Emitted when `value` tokens of token type `id` are
transferred from `from` to `to` by `operator`.
     */
    event TransferSingle(address indexed operator, address indexed
from, address indexed to, uint256 id, uint256 value);

    /**
     * @dev Equivalent to multiple {TransferSingle} events, where
`operator`, `from` and `to` are the same for all
     * transfers.
     */
    event TransferBatch(
        address indexed operator,
        address indexed from,
        address indexed to,
        uint256[] ids,
        uint256[] values
    );

    /**
     * @dev Emitted when `account` grants or revokes permission to
`operator` to transfer their tokens, according to
     * `approved`.
     */
    event ApprovalForAll(address indexed account, address indexed
operator, bool approved);

    /**
     * @dev Emitted when the URI for token type `id` changes to
`value`, if it is a non-programmatic URI.
     *
     * If an {URI} event was emitted for `id`, the standard
     *
https://eips.ethereum.org/EIPS/eip-1155#metadata-extensions[guaran
tees] that `value` will equal the value
     * returned by {IERC1155MetadataURI-uri}.
     */
    event URI(string value, uint256 indexed id);

    /**
     * @dev Returns the amount of tokens of token type `id` owned by `account`.
     *
     * Requirements:
     *
     * - `account` cannot be the zero address.
     */
    function balanceOf(address account, uint256 id) external view returns (uint256);

    /**
     * @dev xref:ROOT:erc1155.adoc#batch-operations[Batched] version of {balanceOf}.
     *
     * Requirements:
     *
     * - `accounts` and `ids` must have the same length.
     */
    function balanceOfBatch(address[] calldata accounts, uint256[] calldata ids)
        external
        view
        returns (uint256[] memory);

    /**
     * @dev Grants or revokes permission to `operator` to transfer the caller's tokens, according to `approved`,
     *
     * Emits an {ApprovalForAll} event.
     *
     * Requirements:
     *
     * - `operator` cannot be the caller.
     */
    function setApprovalForAll(address operator, bool approved) external;

    /**
     * @dev Returns true if `operator` is approved to transfer
```

```
``account``'s tokens.
     *
     * See {setApprovalForAll}.
     */
    function isApprovedForAll(address account, address operator)
external view returns (bool);

    /**
     * @dev Transfers `amount` tokens of token type `id` from
`from` to `to`.
     *
     * Emits a {TransferSingle} event.
     *
     * Requirements:
     *
     * - `to` cannot be the zero address.
     * - If the caller is not `from`, it must have been approved
to spend ``from``'s tokens via {setApprovalForAll}.
     * - `from` must have a balance of tokens of type `id` of at
least `amount`.
     * - If `to` refers to a smart contract, it must implement
{IERC1155Receiver-onERC1155Received} and return the
     * acceptance magic value.
     */
    function safeTransferFrom(
        address from,
        address to,
        uint256 id,
        uint256 amount,
        bytes calldata data
    ) external;

    /**
     * @dev xref:ROOT:erc1155.adoc#batch-operations[Batched]
version of {safeTransferFrom}.
     *
     * Emits a {TransferBatch} event.
     *
     * Requirements:
     *
     * - `ids` and `amounts` must have the same length.
     * - If `to` refers to a smart contract, it must implement
```

```
{IERC1155Receiver-onERC1155BatchReceived} and return the
     * acceptance magic value.
     */
    function safeBatchTransferFrom(
        address from,
        address to,
        uint256[] calldata ids,
        uint256[] calldata amounts,
        bytes calldata data
    ) external;
}
```

### 2. IAgravityNFTMint Contract

```
interface IAgravityNFTMint {

    // Admin use only, ERC1155 allow user own same nft
    function mint(address account, uint256 id, uint256 amount,
bytes memory data) external;

    // Admin use only, ERC1155 allow user own same nft, and admin
call with limited gas
    function mintBatch(address to, uint256[] memory ids, uint256[]
memory amounts, bytes memory data) external;

    // Admin use only, ERC1155 allow user own same nft, and admin
call with limited gas
    function mintBatch(address[] memory accounts, uint256[] memory
ids, uint256[] memory amounts, bytes memory data) external;
}
```

### 3. Agravity NFT

```
contract AgravityNFT is ERC1155Burnable, IAgravityNFTMint,
Ownable, AccessControl, Pausable {
    bytes32 public constant URI_SETTER_ROLE =
keccak256("URI_SETTER_ROLE");
    bytes32 public constant MINTER_ROLE =
keccak256("MINTER_ROLE");
    string public name = "Agravity";

    constructor() ERC1155("https://nft.agravitynft.com/") {
        _setupRole(DEFAULT_ADMIN_ROLE, msg.sender);
        _setupRole(URI_SETTER_ROLE, msg.sender);
        _setupRole(MINTER_ROLE, msg.sender);
    }

    function uri(uint256 _id)public override  view returns (string
memory) {
        string memory id = Strings.toString(_id);
        string memory baseURI = super.uri(_id);
        return string(abi.encodePacked(baseURI,id));
    }

    function setURI(string memory newuri) public
onlyRole(URI_SETTER_ROLE) {
        _setURI(newuri);
    }

    function pause() public onlyOwner {
        _pause();
    }

    function unpause() public onlyOwner {
        _unpause();
    }

    // Admin use only, ERC1155 allow user own same nft
    function mint(address account, uint256 id, uint256 amount,
bytes memory data)
    public
    onlyRole(MINTER_ROLE)
    override
    {
        _mint(account, id, amount, data);
    }

    // Admin use only，ERC1155 allow user own same nft, and admin
call with limited gas
    function mintBatch(address to, uint256[] memory ids, uint256[]
memory amounts, bytes memory data)
    public
    onlyRole(MINTER_ROLE)
    override
    {
        _mintBatch(to, ids, amounts, data);
    }

    // Admin use only，ERC1155 allow user own same nft, and admin
call with limited gas
    function mintBatch(address[] memory accounts, uint256[] memory
ids, uint256[] memory amounts, bytes memory data)
    public
    onlyRole(MINTER_ROLE)
    override
    {
        require(accounts.length == ids.length, "AgravityNFT:
accounts and amounts length mismatch");
        require(ids.length == amounts.length, "AgravityNFT: ids
and amounts length mismatch");

        for (uint256 i = 0; i < accounts.length; i++) {
            _mint(accounts[i], ids[i], amounts[i], data);
        }
    }


    function _beforeTokenTransfer(address operator, address from,
address to, uint256[] memory ids, uint256[] memory amounts, bytes
memory data)
    internal
    whenNotPaused
    override
    {
        super._beforeTokenTransfer(operator, from, to, ids,
amounts, data);
```

```
    }

    function supportsInterface(bytes4 interfaceId)
    public
    view
    override(ERC1155, AccessControl)
    returns (bool)
    {
        return ERC1155.supportsInterface(interfaceId) ||
AccessControl.supportsInterface(interfaceId);
    }
}
```

## READ CONTRACT (ONLY NEED TO KNOW)

1. name

Agravity string

2. owner

0xf330bc9eff5a5006608784f54166c923434e3d54 address

## WRITE CONTRACT

1. burn
account (address)
amount (uint256)
value (uint256)

(The form is filled with the account, amount and value for burn)

2. renounceOwnership
(Renouncing ownership will leave the contract without an owner, thereby removing any functionality that is only available to the owner)

3. transferOwnership
newOwner (address)
(Its function is to change the owner)

## WEBSITE REVIEW

● **Mobile Friendly**

● **Contains no code error**

● **SSL Secured (By R3 SSL)**

**Web-Tech stack:** Next, React, Node js

Domain .com (Eranet) - Tracked by whois

| | |
|---|---|
| First Contentful Paint: | 996ms |
| Fully Loaded Time | 9.6s |
| Performance | 80% |
| Accessibility | 76% |
| Best Practices | 92% |
| SEO | 90% |

## RUG-PULL REVIEW

Based on the available information analyzed by us, we come to the following conclusions:

- 🟢 TOP 5 Holder.

  (*Will be updated after DEX listing*)

- 🟢 The team hasn't done KYC yet.
- 🟢 Owner can mint - (NFT Contract)

## HONEYPOT REVIEW

- 🟢 The owner is not able to pause the contract.
- 🟢 The owner can’t set fees

Note: Please check the disclaimer above and note, that the audit makes no statements or warranties on the business model, investment attractiveness, or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by the project owner.