



**DESSERT**  
FINANCE

## Endless Board Game (ENG)

BEP-20 Audit

Performed at block **19175219**

PERFORMED BY DESSERT FINANCE  
FOR CONTRACT ADDRESS: **0x7EeFB6AeB8BC2c1ba6be1D4273eC0758a1321272**

## INITIAL DISCLAIMER

Dessert Finance provides due-diligence project audits for various projects. Dessert Finance in no way guarantees that a project will not remove liquidity, sell off team supply, or otherwise exit scam.

Dessert Finance does the legwork and provides public information about the project in an easy-to-understand format for the common person.

Agreeing to an audit in no way guarantees that a team will not remove *all* liquidity (“Rug Pull”), remove liquidity slowly, sell off tokens, quit the project, or completely exit scam. There is also no way to prevent private sale holders from selling off their tokens. It is ultimately your responsibility to read through all documentation, social media posts, and contract code of each individual project to draw your own conclusions and set your own risk tolerance.

Dessert Finance in no way takes responsibility for any losses, nor does Dessert Finance encourage any speculative investments. The information provided in this audit is for information purposes only and should not be considered investment advice. Dessert Finance does not endorse, recommend, support, or suggest any projects that have been audited. An audit is an informational report based on our findings, We recommend you do your own research, we will never endorse any project to invest in.

# Table of Contents



1. Contract Code Audit – Token Overview
2. BEP-20 Contract Code Audit – Overview
3. BEP-20 Contract Code Audit – Vulnerabilities Checked
4. Contract Code Audit – Contract Ownership
5. Contract Code Audit – Owner Accessible Functions
6. Liquidity Ownership – Locked / Unlocked
7. Contract Code Audit – Mint Functions
8. Contract Transaction Fees
9. Website Overview
10. Social Media
11. Top Token Holders/Wallets
12. Location Audit
13. Review of Team
14. Roadmap
15. Disclaimers

# Contract Code Audit – Token Overview



# BEP-20 Contract Code Audit – Overview

Dessert Finance was commissioned to perform an audit on Endless Board Game (ENG)

```
Submitted for verification on BscScan.com on 2022-09-06
// Title: https://github.com/centipede44/contracts/blob/master/contracts/221/Address.sol

pragma solidity ^4.4.0

/**
 * @dev Collection of functions related to the address type
 */
library Address {
    /**
     * @dev Returns true if `account` is a contract.
     *
     * @param account Address
     *
     * @return bool
     *
     * @dev This is useful to avoid trying to transfer to an account (e.g. a contract)
     * that is not a contract (e.g. a contract address).
     *
     * @dev Beware that this returns false for the following
     * types of addresses:
     *
     * - an externally-owned account
     * - a contract in construction
     * - an address where a contract will be created
     * - an address where a contract lived, but was destroyed
     */
    function isContract(address account) internal view returns (bool) {
        // According to EIP-1052, this is the value returned for non-existent accounts
        // and the value returned for accounts that have been deleted.
        // For accounts without code, i.e. "externally-owned"
        // https://etherscan.io/address/0x0000000000000000000000000000000000000000
        if (account == 0x0000000000000000000000000000000000000000) {
            return false;
        }
        // If we have the address data, we can check if there is any code.
        return !account.isEqual(0x0000000000000000000000000000000000000000);
    }

    /**
     * @dev Replacement for Solidity's `transfer()` sending `amount` wei to
     * `recipient`. Forwarding all available gas and forwarding on errors.
     *
     * https://etherscan.io/address/0x0000000000000000000000000000000000000000 increases the gas cost
     * of certain transfers, possibly making contracts go over the 100k gas limit
     * bypassed by `transfer`, using this instead to reduce gas cost
     * `transfer` (unlike) reserves this limitation
     * https://etherscan.io/address/0x0000000000000000000000000000000000000000
     *
     * @param recipient Address
     * @param amount uint256
     *
     * @return bool
     *
     * @dev Note that because control is transferred to `recipient`, care must be
     * taken to not create reentrancy vulnerabilities. Consider using
     * `ReentrancyGuard` on the
     * https://etherscan.io/address/0x0000000000000000000000000000000000000000
     * https://etherscan.io/address/0x0000000000000000000000000000000000000000
     */
    function sendEther(address payable recipient, uint256 amount) internal {
        require(address(this).balance > amount, "Address: insufficient balance");

        // Similar to Etherbase, we use the `value` field of the call, which will value
        // (and success) to the recipient call value.
        recipient.call{value: amount}("");
        require(recipient.balance >= amount, "Address: unable to send value, recipient may have reverted");
    }

    /**
     * @dev Performs a Solidity function call using a low level `call`.
     *
     * @param target Address
     * @param data bytes
     *
     * @return bool
     *
     * @dev If `target` reverts with a revert reason, use the data returned by this
     * function (like regular Solidity function calls).
     *
     * @dev Returns the raw returned data. To convert to the expected return value,
     * see https://etherscan.io/address/0x0000000000000000000000000000000000000000
     * https://etherscan.io/address/0x0000000000000000000000000000000000000000
     */
    function call(address target, bytes data) internal returns (bool, bytes) {
        require(target.isContract(), "Address: call to non-contract");
        require(target.balance > 0, "Address: no gas");
        (bool success, bytes data) = target.call{gas: gasleft(), value: 0}(data);
        return (success, data);
    }
}

// Requirements:
// - `target` must be a contract.
// - calling `target` with `data` must not revert.
```

## Contract Address

0x7EeFB6AeB8BC2c1ba6be1D4273eC0758a1321272

## TokenTracker

Endless Board Game (ENG)

## Contract Creator

0xf8596adD93fE544F9c04e9934fA9096Aa38be1b2

## Source Code

Contract Source Code Verified

## Contract Name

Token

## Other Settings

default evmVersion, MIT

## Compiler Version

v0.6.2+commit.bacdbe57

## Optimization Enabled

No with 200 runs

Code is truncated to fit the constraints of this document.

[The code in its entirety can be viewed here.](#)

The contract code is **verified** on BSCScan.

# BEP-20 Contract Code Audit – Vulnerabilities Checked

Vulnerability Tested	AI Scan	Human Review	Result
Compiler Errors	Complete	Complete	✓ Low / No Risk
Outdated Compiler Version	Complete	Complete	✓ Low / No Risk
Integer Overflow	Complete	Complete	✓ Low / No Risk
Integer Underflow	Complete	Complete	✓ Low / No Risk
Correct Token Standards Implementation	Complete	Complete	✓ Low / No Risk
Timestamp Dependency for Crucial Functions	Complete	Complete	✓ Low / No Risk
Exposed _Transfer Function	Complete	Complete	✓ Low / No Risk
Transaction-Ordering Dependency	Complete	Complete	✓ Low / No Risk
Unchecked Call Return Variable	Complete	Complete	✓ Low / No Risk
Use of Deprecated Functions	Complete	Complete	✓ Low / No Risk
Unprotected SELFDESTRUCT Instruction	Complete	Complete	✓ Low / No Risk
State Variable Default Visibility	Complete	Complete	✓ Low / No Risk
Deployer Can Access User Funds	Complete	Complete	✓ Low / No Risk

The contract code is verified on BSCScan.

The vulnerabilities listed above were not found in the token's Smart Contract.

# Contract Code Audit – Contract Ownership

This is not an Ownable Contract



This is not an Ownable contract.

# Liquidity Ownership – Locked / Unlocked

No locked liquidity information has been found.



This page will contain links to locked liquidity for the project if we are able to locate that information. Locked liquidity information was not found on the project's website.



# Contract Code Audit – Mint Functions

This Contract Cannot Mint ENGTokens.

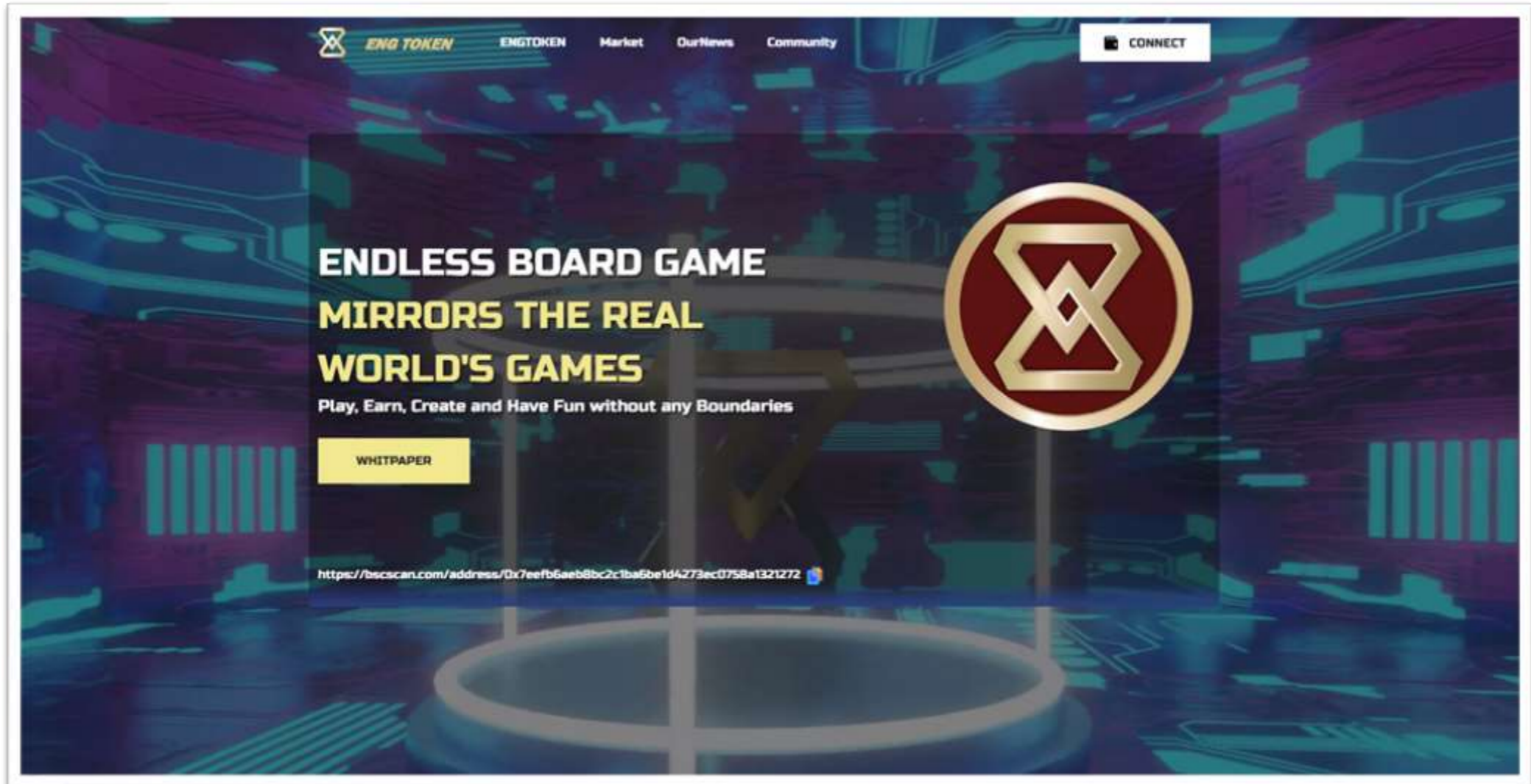


We do understand that sometimes mint functions are essential to the functionality of the project.

**A mint function was not found in the contract code.**

# Website Part 1 – Overview

[www.endlessboardgame.com](http://www.endlessboardgame.com)



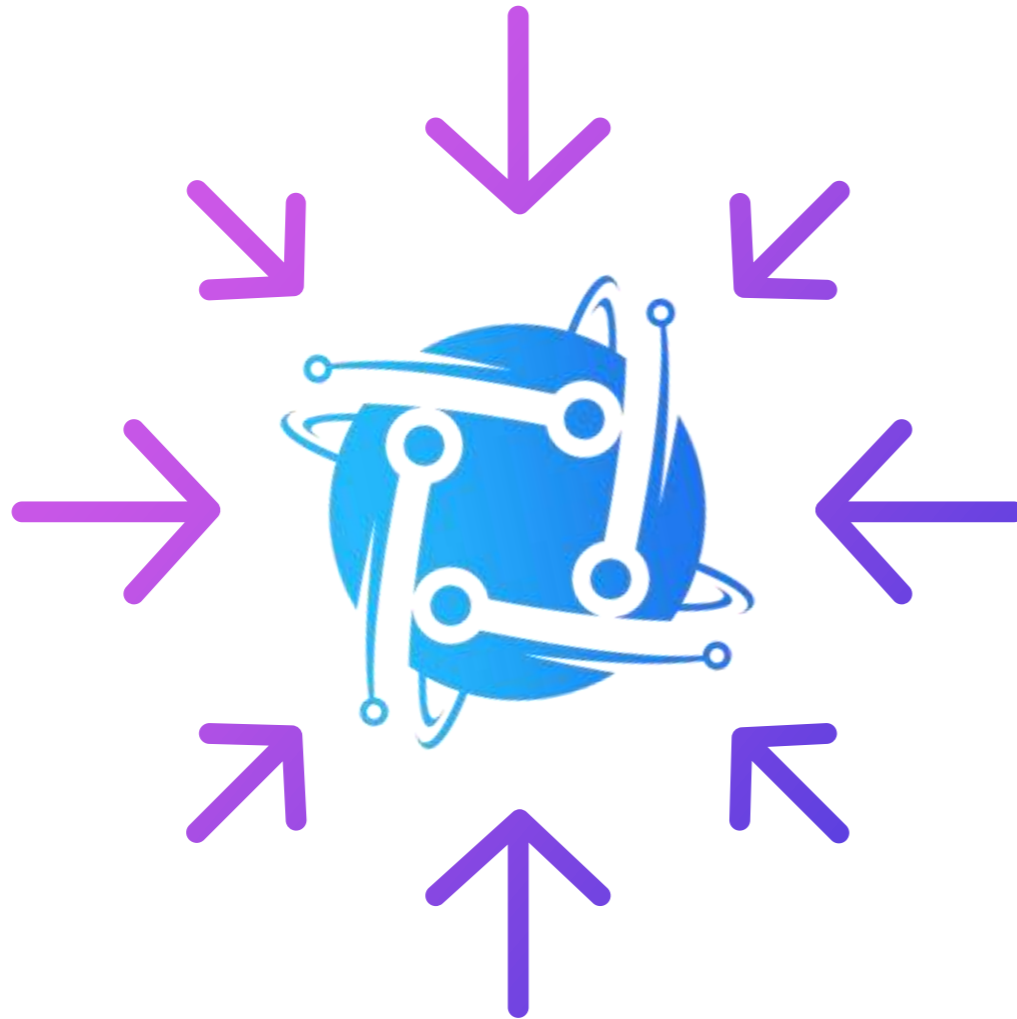
Above images are actual snapshots of the current live website of the project.

Website was registered on 08/01/2022, registration expires 08/01/2023.

**X** This does not meet the 3 year minimum we like to see on new projects.



## Website Part 2 – Checklist



- ✓ Mobile Friendly
- ✓ No JavaScript Errors
- ✓ Spell Check
- ✓ SSL Certificate

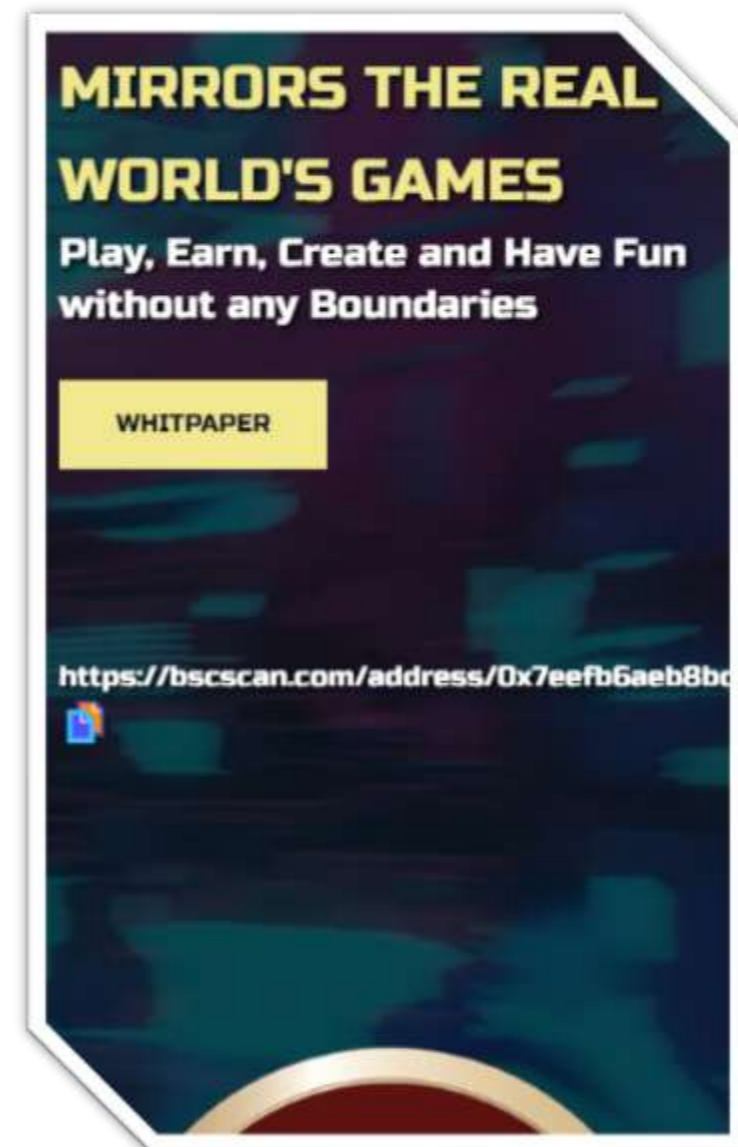
The website contained no JavaScript errors. No typos, or grammatical errors were present, and we found a valid SSL certificate allowing for access via https.

No additional issues were found on the website.

# Website Part 3 – Responsive HTML5 & CSS3

No issues were found on the Mobile Friendly check for the website. All elements loaded properly and browser resize was not an issue. The team has put a considerable amount of thought and effort into making sure their website looks great on all screens.

No severe JavaScript errors were found. No issues with loading elements, code, or stylesheets.



# Website Part 4 (GWS) – General Web Security



## SSL CERTIFICATE

A valid SSL certificate was found. Details are as follows:

Offered to: endlessboardgame.com

Issued by: Cloudflare Inc

Valid Until: 06/20/2023



## CONTACT EMAIL

A valid contact email was found on the official website. Contact email is listed as shown below:

[Contact](#)

**info@endlessboardgame.com**

engtokenpro@gmail.com



## SPAM / MALWARE / POPUPS

No malware found

No injected spam found

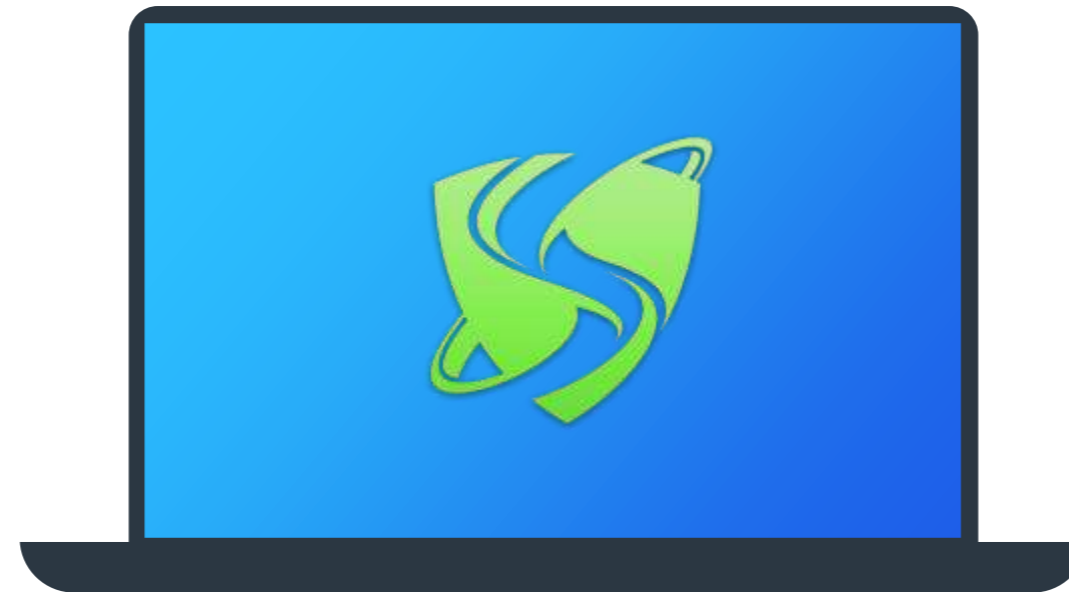
No internal server errors

No popups found

Domain is marked clean by Google, McAfee, Sucuri Labs, & ESET



# Social Media



We were able to locate a variety of Social Media networks for the project.

All links have been conveniently placed below.



[Twitter](#)



[Telegram](#)



[Reddit](#)



[Facebook](#)



[Instagram](#)

✓ **At least 3 social media networks were found.**

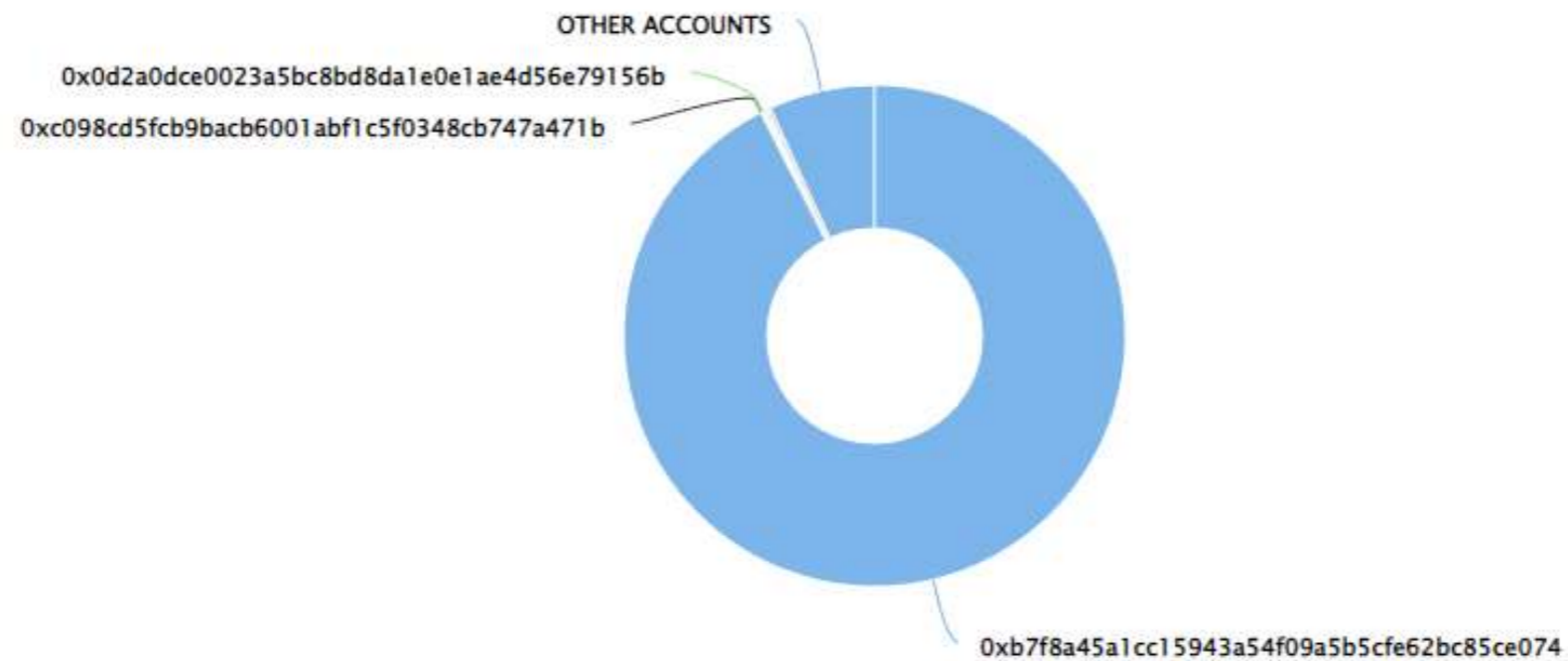
# Top Token Holders

The top token holders at the time of the audit are shown below.

[Click here to view the most up-to-date list of holders](#)

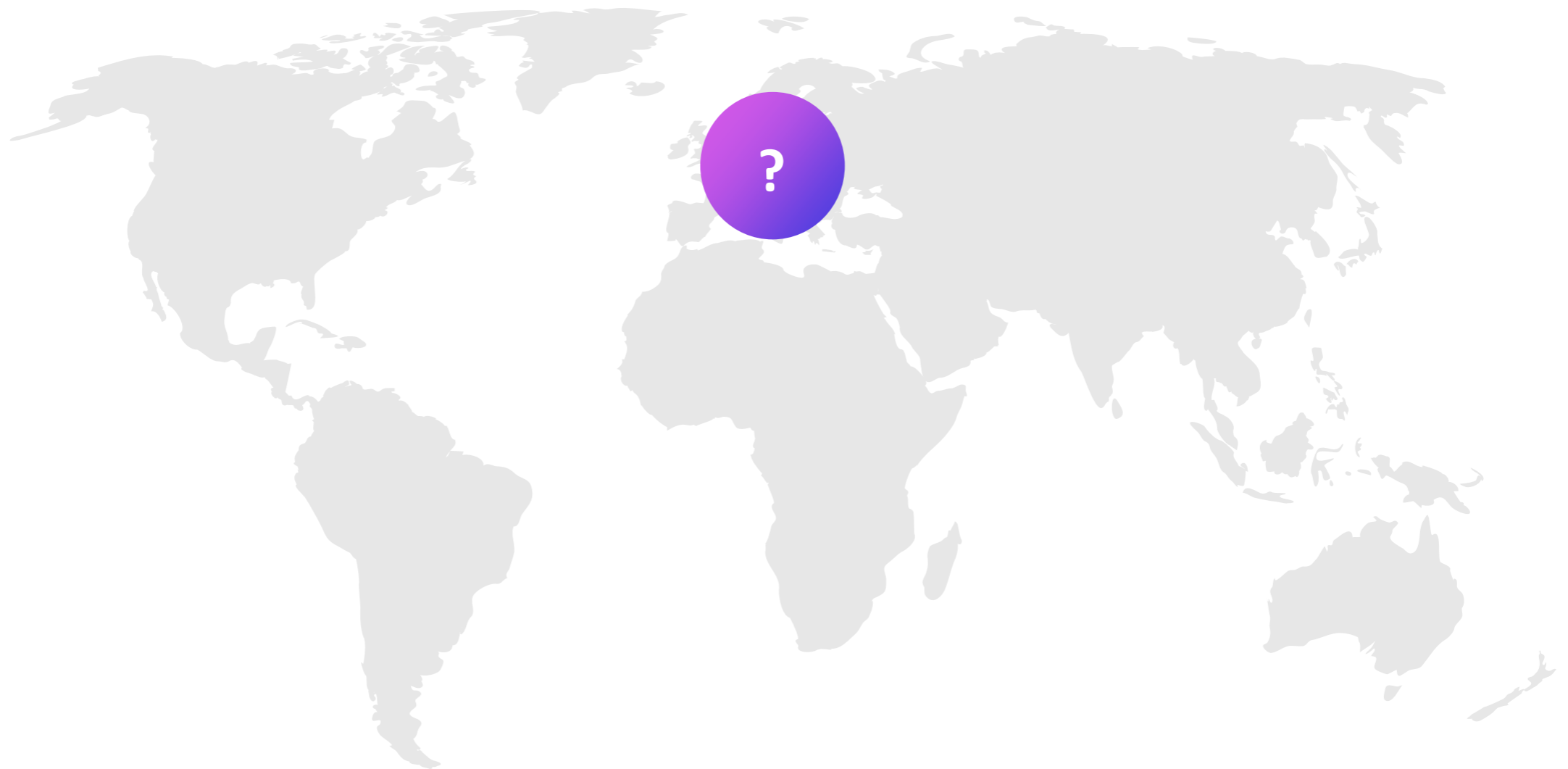
## Endless Board Game Top 100 Token Holders

Source: BscScan.com



# Location Audit

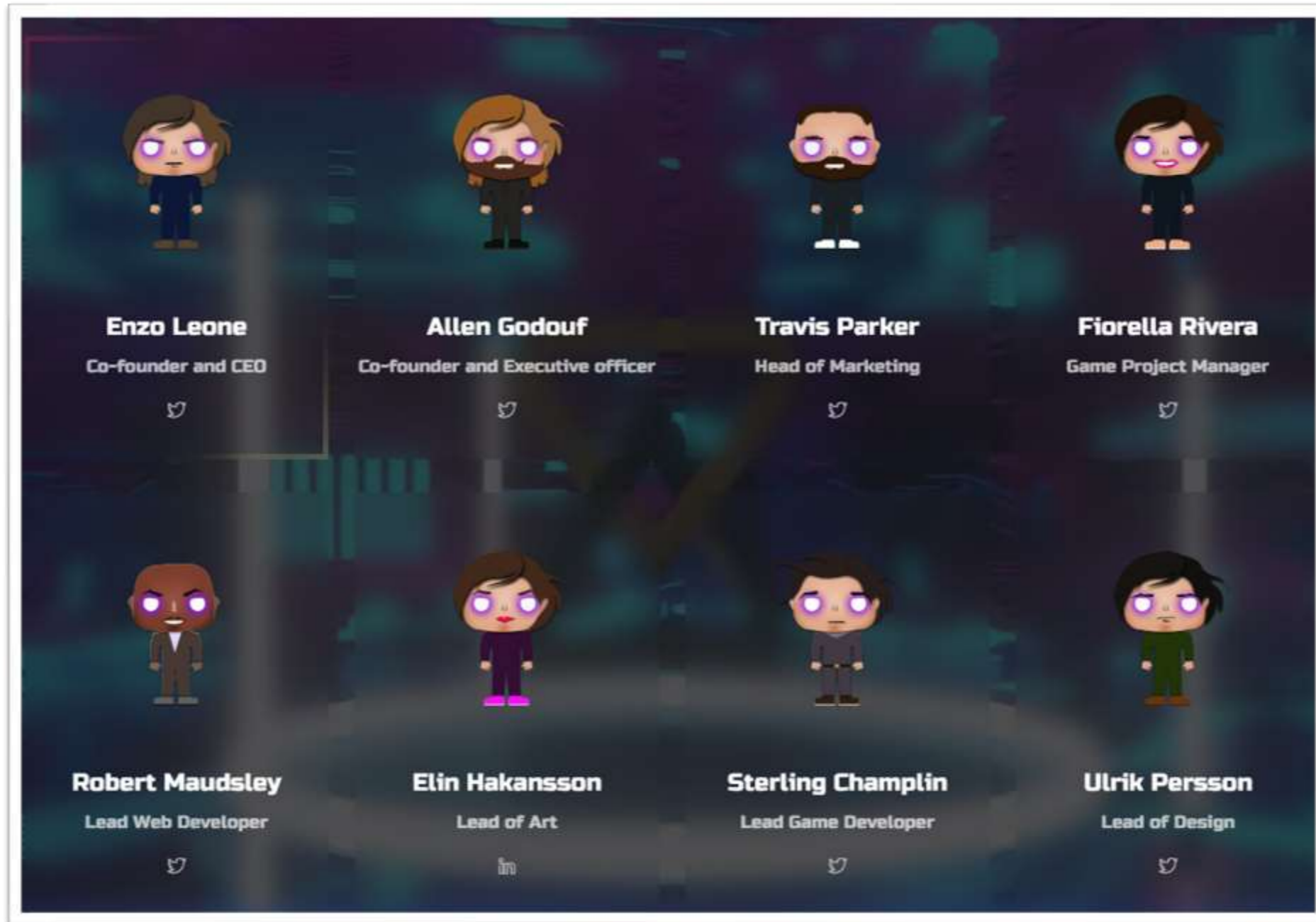
We were unable to identify a primary location for the project at this time or a location has not been declared.





# Team Overview

The following information regarding the team was found on the projects website.



# Roadmap

*A roadmap was found on the official website, we have conveniently placed it on this page for your viewing.*



# Disclaimer



The opinions expressed in this document are for general informational purposes only and are **not intended to provide specific advice or recommendations for any individual or on any specific investment**. It is only intended to provide education and public knowledge regarding projects. This audit is only applied to the type of auditing specified in this report and the scope of given in the results. Other unknown security vulnerabilities are beyond responsibility. Dessert Finance only issues this report based on the attacks or vulnerabilities that already existed or occurred before the issuance of this report. For the emergence of new attacks or vulnerabilities that exist or occur in the future, Dessert Finance lacks the capability to judge its possible impact on the security status of smart contracts, thus taking no responsibility for them. The smart contract analysis and other contents of this report are based solely on the documents and materials that the contract provider has provided to Dessert Finance or was publicly available before the issuance of this report (issuance of report recorded via block number on cover page), if the documents and materials provided by the contract provider are missing, tampered, deleted, concealed or reflected in a situation that is inconsistent with the actual situation, or if the documents and materials provided are changed after the issuance of this report, Dessert Finance assumes no responsibility for the resulting loss or adverse effects. Due to the technical limitations of any organization, this report conducted by Dessert Finance still has the possibility that the entire risk cannot be completely detected. Dessert Finance disclaims any liability for the resulting losses.

Dessert Finance provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Even projects with a low risk score have been known to pull liquidity, sell all team tokens, or exit-scam. Please exercise caution when dealing with any cryptocurrency related platforms.

The final interpretation of this statement belongs to Dessert Finance.

Dessert Finance highly advises against using cryptocurrencies as speculative investments and they should be used solely for the utility they aim to provide.



# Thank You

DESSERT FINANCE PROJECT AUDIT HAS BEEN COMPLETED FOR ENDLESS BOARD GAME (ENG) 1 DSRT HAS BEEN SENT TO AUDITED PROJECT'S CONTRACT ADDRESS FOR VERIFICATION OF THIS AUDIT AT BLOCK NUMBER: **19175219**

**THIS AUDIT IS ONLY VALID IF VIEWED ON [HTTPS://WWW.DSSERTSWAP.FINANCE](https://www.dessertswap.finance)**

[www.dessertswap.finance](https://www.dessertswap.finance)  
<https://t.me/dessertswap>