August 2021

Grogu AUDIT

BEP-20 TOKEN

Prepared by: Coinscope team

For Contract Address (testnet):
0x9BF44d9e8D9Ae56197068eAB2d2c0572996C3ef7

Table of Contents

Contract Review

Contract Name	GRGU_v35
Compiler Version	v0.8.6+commit.11564f7e
Optimization	200 runs
Licence	GNU GPLv3 license

Audit Updates

Initial Audit	11/08/2021
Corrected	25/08/2021

PI - Performance Improvement

Criticality	low
Location	https://testnet.bscscan.com/address/0x9BF44d9e8D9Ae56197068eAB2d2c0572996C3ef7#code#L92
Status	**Resolve** Grogu team response: *This is a universal library pulled into all our contracts -- it's only flattened on deployment for readability's sake -- so we cannot separate out the hashing function. In the library it'd have to be a struct and then called in the function, costing more bytecode and gas, so it was left alone.*

Description

The static assignment could be defined outside the function scope. The assignment creates one extra execution statement. Usually, the compiler will optimize it, but it is better not to rely on this.

```
bytes32 accountHash =
0xc5d2460186f7233c927e7db2dcc703c0e500b653ca82273b7bfad8045d85a470
;
```

Recommendation

The variable could be defined in the class scope, so the L92 could be eliminated.

```
bytes32 private constant accountHash =
0xc5d2460186f7233c927e7db2dcc703c0e500b653ca82273b7bfad8045d85a470
;
```

RC - REDUNDANT CODE

Criticality	low
Location	https://testnet.bscscan.com/address/0x9BF44d9e8D9Ae56197068eAB2d2c0572996C3ef7#code#L417
Status	**Resolved**

Description

The function sets the ownership according to the argument's address. After the assignment, an event is triggered that propagates the previous and the next owner.

```
function _setOwner(address newOwner) private {
    address oldOwner = _owner;
    _owner = newOwner;
    emit OwnershipTransferred(oldOwner, newOwner);
}
```

Recommendation

Since there is not any concurrency issue, the event could be triggered first.

```
function _setOwner(address newOwner) private {
    emit OwnershipTransferred(_owner, newOwner);
    _owner = newOwner;
}
```

RC - REDUNDANT CODE

Criticality	low
Location	https://testnet.bscscan.com/address/0x9BF44d9e8D9Ae56197068eAB2d2c0572996C3ef7#code#L423
Status	**Resolved**

Description

The function sets the operator according to the argument's address. After the assignment, an event is triggered that propagates the previous and the next operator.

```
function _setOperator(address newOperator) private {
    address oldOperator = _operator;
    _operator = newOperator;
    emit OwnershipTransferred(oldOperator, newOperator);
}
```

Recommendation

Since there is not any concurrency issue, the event could be triggered first.

```
function _setOperator(address newOperator) private {
    emit OwnershipTransferred(_operator, newOperator);
    _operator = newOperator;
}
```

RC - REDUNDANT CODE

Criticality	low
Location	https://testnet.bscscan.com/address/0x9BF44d9e8D9Ae56197068eAB2d2c0572996C3ef7#code#L429
Status	**Resolved**

Description

The function sets the multiSig according to the argument's address. After the assignment, an event is triggered that propagates the previous and the next multiSig.

```
function _setMultiSig(address newMultiSig) private {
    address oldMultiSig = _multisig;
    _multisig = newMultiSig;
    emit MultiSigTransferred(oldMultiSig, newMultiSig);
}
```

Recommendation

Since there is not any concurrency issue, the event could be triggered first.

```
function _setMultiSig(address newMultiSig) private {
    emit MultiSigTransferred(_multisig, newMultiSig);
    _multisig = newMultiSig;
}
```

MS - MISSING STATEMENT

Criticality	low
Location	https://testnet.bscscan.com/address/0x9BF44d9e8D9Ae56197068eAB2d2c0572996C3ef7#code#L472
Status	**Resolved**

Description

The mint wrapper toggles the mint functionality. It is an essential part of the contract features. Hence, the mint event should be triggered.

```
function _start_mint_wrappers() internal virtual
whenMintWrappersOff {
    _mint_wrappers_stopped = false;
    //emit Started_Mint_Wrappers(_msgSender());
}
```

Recommendation

Remove the code comment that triggers the “mint started” event.

```
function _start_mint_wrappers() internal virtual
whenMintWrappersOff {
    _mint_wrappers_stopped = false;
    emit Started_Mint_Wrappers(_msgSender());
}
```

MS - MISSING STATEMENT

Criticality	low
Location	https://testnet.bscscan.com/address/0x9BF44d9e8D9Ae56197068eAB2d2c0572996C3ef7#code#L786
Status	**Resolved**

Description

The nocontracts variation enables and disables all the transfer operations in the contract. It is used in the `_transfer` function. Since it is so critical for the contract operation, the disabled event should be notified.

```
function _start_nocontracts() internal virtual whenNoContractsOff
{
    _nocontracts_stopped = false;
    // emit Started_NoContracts(_msgSender());
}
```

Recommendation

Remove the code comment that triggers the “mint started” event.

```
function _start_nocontracts() internal virtual whenNoContractsOff
{
    _nocontracts_stopped = false;
    emit Started_NoContracts(_msgSender());
}
```

IO - Check for Integer Overflow

Criticality	low
Location	https://testnet.bscscan.com/address/0x9BF44d9e8D9Ae56197068eAB2d2c0572996C3ef7#code#L1666
Status	**Resolve** Grogu team response: *SafeMath is no longer needed, the header comments and lines 1350ish to 1355 link to the solidity docs and further explanations. In short, in solc ^0.8.0 it'll revert on over/underflow by default, calling it "unchecked" would save gas but also allow the previous behavior of needing SafeMath wrappers resulting in more gas needed per TX. So it was left alone.*

Description

The `_totalSupply` is an `uint256`. The amount is also an `uint256`. The `_balances` a hash map that points `uint256` from `address`. Those two statements do not check for potential integer overflow.

```
function _mint(address account, uint256 amount) internal virtual {
    require(account != address(0), "mint to zero address");

    _beforeTokenTransfer(address(0), account, amount);

    _totalSupply += amount;
    _balances[account] += amount;
    emit Transfer(address(0), account, amount);

    _afterTokenTransfer(address(0), account, amount);
}
```

Recommendation

We advise the client to use a mathematical library that handles this kind of issues, like the `SafeMath` library of `Openzeppelin` library. Otherwise they could manually check if the next sum reaches the `uint256` limit.

```
function _mint(address account, uint256 amount) internal virtual {
    require(account != address(0), "mint to zero address");

    _beforeTokenTransfer(address(0), account, amount);

    _totalSupply = _totalSupply.add(amount);
    _balances[account] = _balances[account].add(amount);
    emit Transfer(address(0), account, amount);

    _afterTokenTransfer(address(0), account, amount);
}
```

MN - Misleading Name

Criticality	low
Location	https://testnet.bscscan.com/address/0x9BF44d9e8D9Ae56197068eAB2d2c0572996C3ef7#code#L1755
Status	**Resolve** Grogu team response: *removed completely. We're not and won't oversupply, and won't then need to "burn"*

Description

The _charity function and the comments above this function, gives the perspective that this function donates a specific amount to the charity address. This is not happening inside the function. It merely removes the amount from the account balance and the total supply.

```
function _charity(address account, uint256 amount) internal
virtual {
    require(account != address(0), "charity from the 0 address");

    _beforeTokenTransfer(account, address(0), amount);

    uint256 accountBalance = _balances[account];
    require(accountBalance >= amount, "charity amount exceeds
bal");
    unchecked {
        _balances[account] = accountBalance - amount;
    }
    _totalSupply -= amount;

    emit Transfer(account, address(0), amount);

    _afterTokenTransfer(account, address(0), amount);
}
```

Recommendation

If the client needs this functionality, then the function should change its name to something more relevant like “_unmint”. Otherwise, the client should add the corresponding charity transfer functionality.

MFC - Multiple Function Calls

Criticality	low
Location	https://testnet.bscscan.com/address/0x9BF44d9e8D9Ae56197068eAB2d2c0572996C3ef7#code#L1869
Status	Resolve Grogu team response: *Most of the times the first check should stop the loop, and declaring another variable would increase bytecode -- which is an issue with this large contract*

Description

The function maxTransferAmount() is called twice in the antiWhale modifier despite the fact that the result does not change in the current execution thread. The maxTransferAmount() is not just a getter, it calculates the number.

```
modifier antiWhale(
    address sender,
    address recipient,
    uint256 amount
) {
    if (maxTransferAmount() > 0) {
        if (
            _excludedFromAntiWhale[sender] == false &&
            _excludedFromAntiWhale[recipient] == false
        ) {
            require(
                amount <= maxTransferAmount(),
                "antiWhale::exceeds maxTXAmount"
            );
        }
    }
    _;
}
```

Recommendation

The client could avoid the duplication and call this function once. It is discussable since there is a balance between the extra statements and the gas that is required for the operation.

```
modifier antiWhale(
    address sender,
    address recipient,
    uint256 amount
) {
    uint256 maxAmount = maxTransferAmount();
    if (maxAmount > 0) {
        if (
            _excludedFromAntiWhale[sender] == false &&
            _excludedFromAntiWhale[recipient] == false
        ) {
            require(
                amount <= maxAmount,
                "antiWhale::exceeds maxTXAmount"
            );
        }
    }
    _;
}
```

PAP - Public Access Permissions

Criticality	low
Location	https://testnet.bscscan.com/address/0x9BF44d9e8D9Ae56197068eAB2d2c0572996C3ef7#code#L2114
Status	**Resolved**

Description

The `F4_isBlacklisted` is a getter that yields the blacklisted accounts. It is quite helpful information that could be exposed to the public without limitations. Currently it can only be called from the `operator`.

```
function F4_isBlacklisted(address _account) public view
onlyOperator returns (bool) {
    return _blacklist[_account];
}
```

Recommendation

The restriction of the role access could be removed. Hence, the functions could be available to the public.

```
function F4_isBlacklisted(address _account) public view returns
(bool) {
    return _blacklist[_account];
}
```

PAP - Public Access Permissions

Criticality	low
Location	https://testnet.bscscan.com/address/0x9BF44d9e8D9Ae56197068eAB2d2c0572996C3ef7#code#L2114
Status	**Resolved**

Description

The `G4_isExcludedFromAntiWhale` is a getter that describes if an account is excluded from the anti-whale mechanism. It is quite helpful information that could be exposed to the public without limitations. Currently it can only be called from the `operator`.

```
function G4_isExcludedFromAntiWhale(address _account) public view
onlyOperator returns (bool) {
    return _excludedFromAntiWhale[_account];
}
```

Recommendation

The restriction of the role access could be removed. Hence, the functions could be available to the public.

```
function G4_isExcludedFromAntiWhale(address _account) public view
returns (bool) {
    return _excludedFromAntiWhale[_account];
}
```

Community-Controlled Multi-Signature Model

Grogu introduces a novel way to control the administration action that can be executed from the contract. The governor-voting pattern is not new in computer science, it is a well-known pattern that assists in choosing the decision-maker. Grogu introduces this pattern to smart contract technology.

Roles

The administration is separated in 3 roles:

- The owner
- The operator
- The multiSig

Each of the roles is responsible for mutating a specific group of functions in the contract. The following is the set of flags that every role is responsible for.

Role	Flag
operator	mint_wrappers_stopped
multiSig	change_operator_stopped
multiSig	rate_change_stopped

multiSig	blacklist_stopped
multiSig	antiwhale_stopped
operator	owner_privileges_stopped
multiSig	approve_spendable_address_stopped
operator	nocontracts_stopped
multiSig	multisig_stopped

Future Work

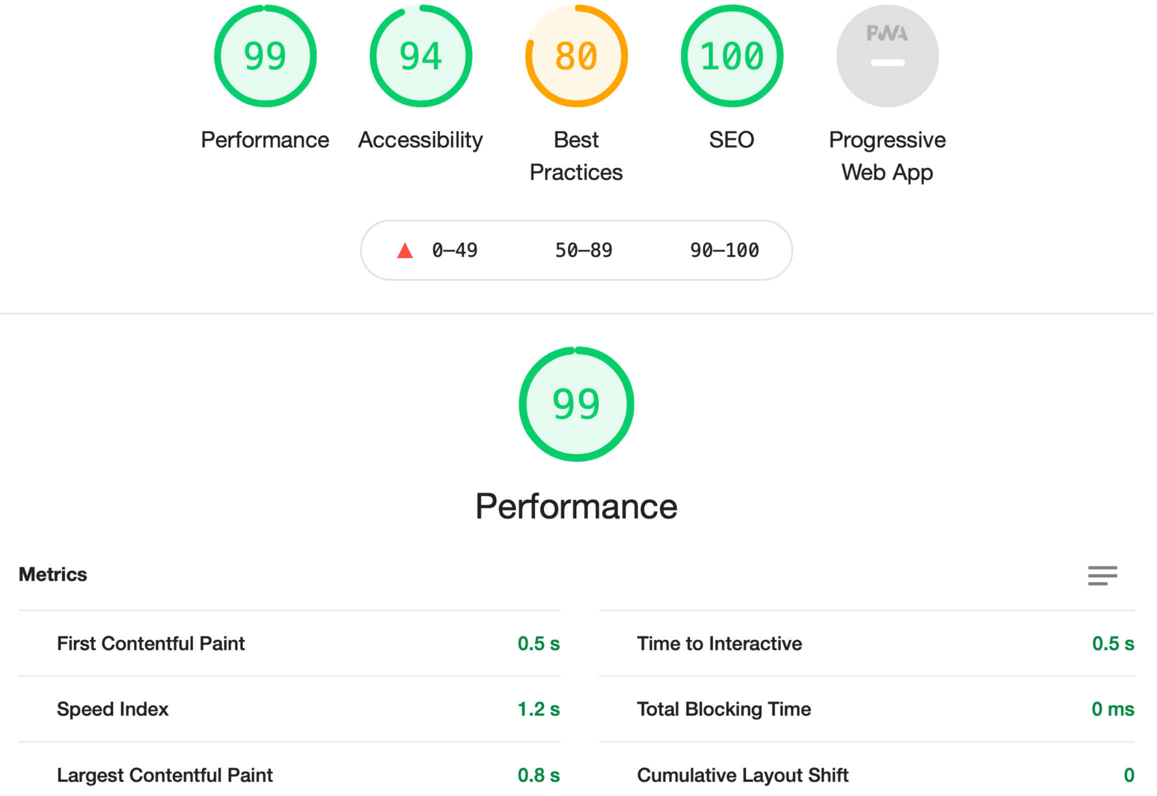
The project sets the fundamentals of a voting-pattern ecosystem. It could potentially be baked more inside the transaction functions. Even if we observe this approach inside the contract code, it would be interesting to see how it would operate in other circumstances. For instance, allowing to execute a mutable function only if the caller has gathered a specific percentage of votes.

Comment

The multi-signature token model, as it is implemented in the Grogu contract, does not guarantee that administrators are not able to harm the inner state of the contract. When the operator or the multiSig are taking the permissions, they are free to execute the functions at their own will.

Website Diagnostics

Desktop



99 Performance

94 Accessibility

80 Best Practices

100 SEO

PWA Progressive Web App

▲ 0–49 50–89 90–100

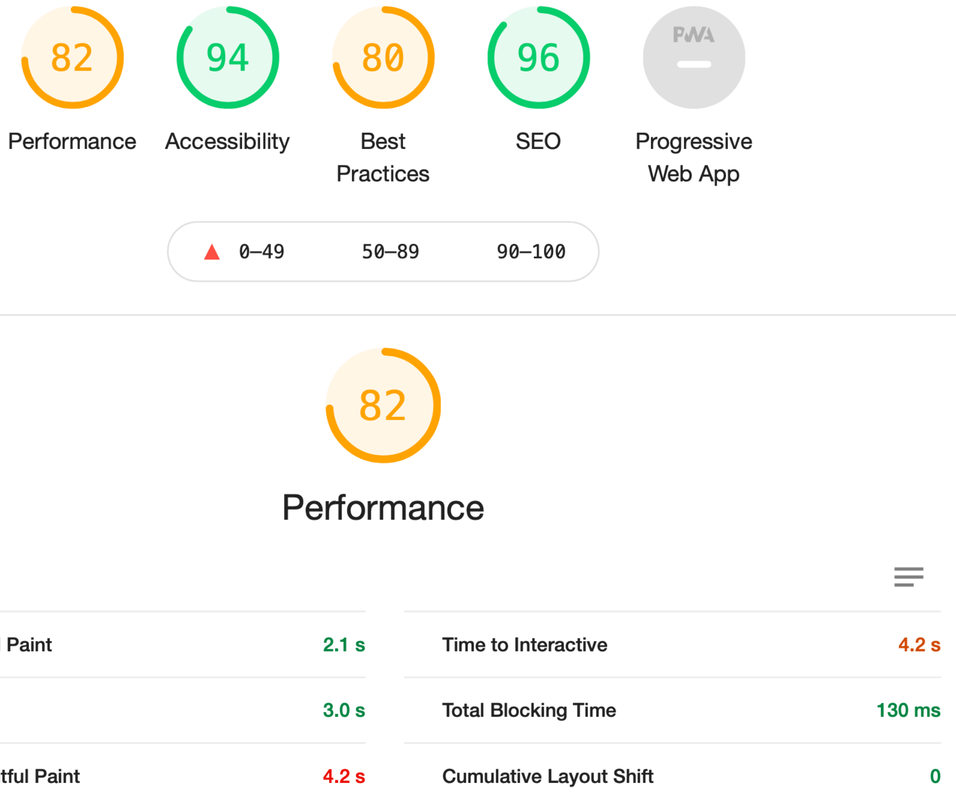
99

Performance

Metrics

First Contentful Paint	0.5 s	Time to Interactive	0.5 s
Speed Index	1.2 s	Total Blocking Time	0 ms
Largest Contentful Paint	0.8 s	Cumulative Layout Shift	0

Mobile



82 Performance

94 Accessibility

80 Best Practices

96 SEO

Progressive Web App

▲ 0–49 50–89 90–100

82

Performance

Metrics

First Contentful Paint	2.1 s	Time to Interactive	4.2 s
Speed Index	3.0 s	Total Blocking Time	130 ms
▲ Largest Contentful Paint	4.2 s	Cumulative Layout Shift	0

Report

We are using the Absolute Category Rating (ACR) in order to measure the quality.

The levels of the scale are, sorted by quality in decreasing order:

5 Excellent; 4 Very Good; 3 Good; 2 Fair; 1 Poor;

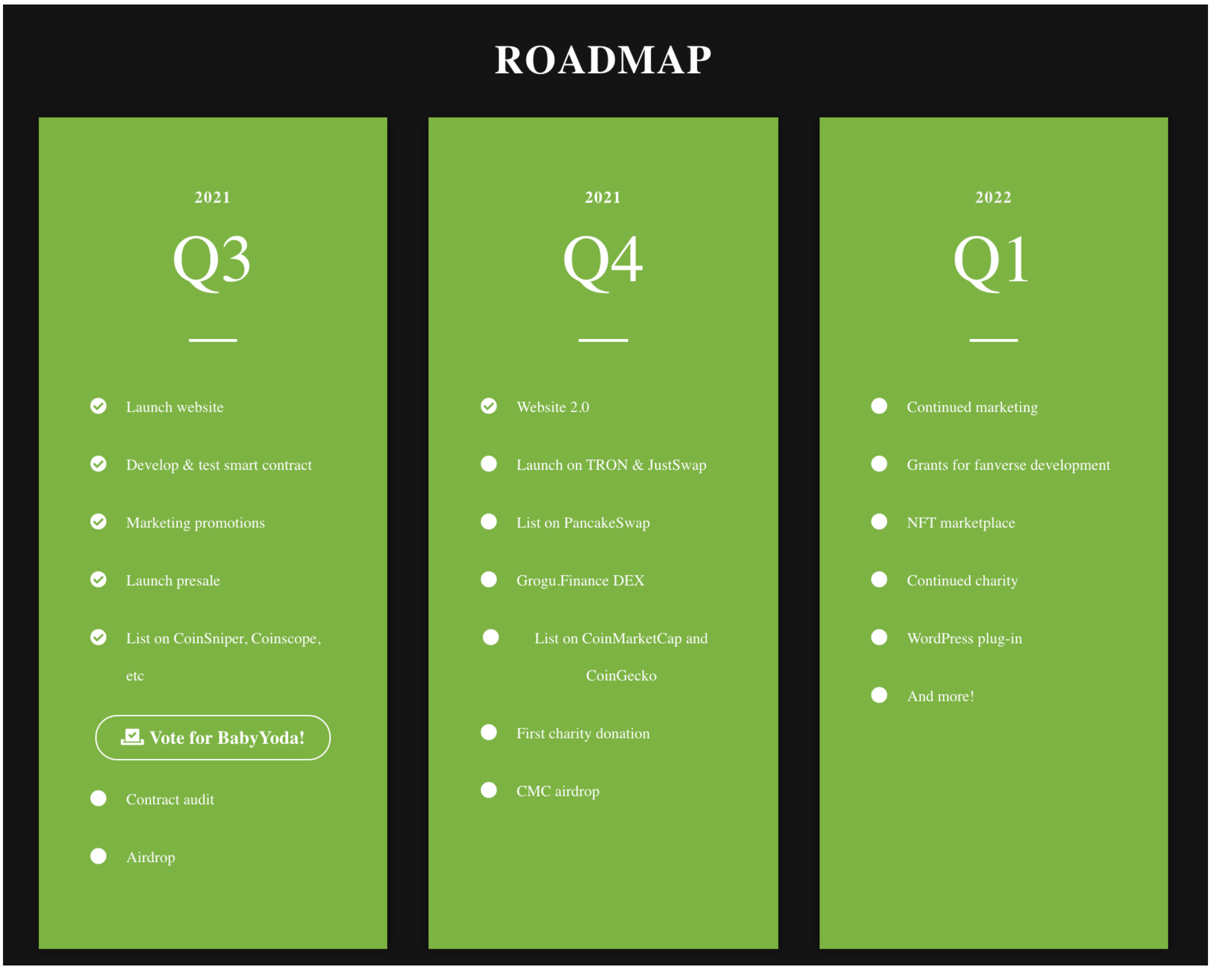
Performance	Excellent
Best Practices	Very Good
Accessibility	Excellent
SEO ranking	Excellent

Comment

The website is performing well. There are some minor improvements that could be implemented in order to reach the 100% rank.

- Background and foreground colors do not have a sufficient contrast ratio.
- Heading elements are not in a sequentially-descending order
- Links do not have a discernible name
- Does not use HTTPS — 1 insecure request found
- Links to cross-origin destinations are unsafe

Roadmap



ROADMAP

2021

Q3

- ✓ Launch website
- ✓ Develop & test smart contract
- ✓ Marketing promotions
- ✓ Launch presale
- ✓ List on CoinSniper, Coinscope, etc

Vote for BabyYoda!

- ● Contract audit
- ● Airdrop

2021

Q4

- ✓ Website 2.0
- ● Launch on TRON & JustSwap
- ● List on PancakeSwap
- ● Grogu.Finance DEX
- ● List on CoinMarketCap and CoinGecko
- ● First charity donation
- ● CMC airdrop

2022

Q1

- ● Continued marketing
- ● Grants for fanverse development
- ● NFT marketplace
- ● Continued charity
- ● WordPress plug-in
- ● And more!

Grogu seems to reach the roadmap target for the Q3 of 2021. Almost all the bullets have been achieved. In the Q4 of 2021 there are 2 targets that could potentially raise Grogu popularity.

- Launching on different network chains
- Charity donations.

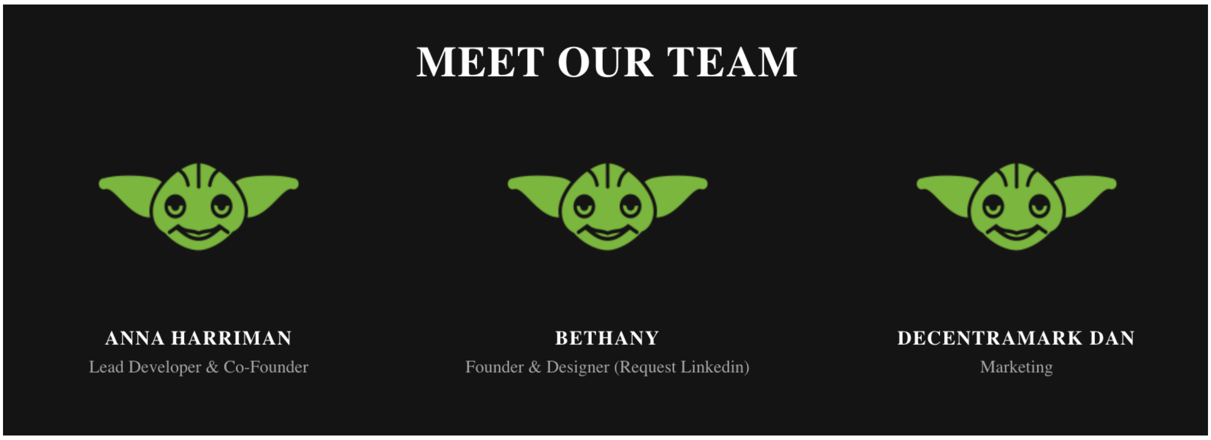
Charity donations are usually appreciated by the community. Some potential issues for the Q4 is the target of:

- List on CoinMarketCap
- List on CoinGecko

Update

The Grogu schedule seems to be on time. They have added 2 games in Google Play and are trying to make their trademark unique.

Team



MEET OUR TEAM

ANNA HARRIMAN
Lead Developer & Co-Founder

BETHANY
Founder & Designer (Request Linkedin)

DECENTRAMARK DAN
Marketing

The team has 3 people visible on the website. Anna Harriman is linked to Linkedin. The profile seems quite idle in regards with the age of experience that it refers to. Bethany does not have any physical existence in social profiles. It is linked to the website’s email address. Decentramark Dan is not a person but an organization that handles decentralised marketing.

Update

The Grogu team has provided to Coinscope all the identification information. KYC is verified and publicly linked on grogu.finance. Ee have a copy of their current and valid First-World Incorporation papers and federal tax id.

Tokenomics

The contract’s tax, burn, donation and reflection rates are mutable so we cannot extract a stable list of percentages. Grogu states the following:

- Fair launch/zero issuance
- Initial emission to airdrop participants only, then to supply tokens for the presale
- Pre-sale of 2,500,000 tokens @ $0.025 each.
- 100% of the presale will provide permanently locked liquidity
- CoinMarketCap airdrop of 1,000,000
- Airdrop to pre-sale investors of 100,000
- Marketers will be rewarded with 700,000 tokens over three years. Each allotment locked for one year.
- Content/platform moderator allocation 700,000
- Charity Fund: Locked in a multi-sig wallet, one key to the dev team, one to the community: 3,300,000
- IDO (Initial DEX Offering) – 10,000,000
- Treasury Wallet to fund future development 6,200,000
- The pre-sale will permanently lock $125,000 in Grogu/BNB and Grogu/BUSD.
- A successful IDO will raise $250,000, less fees. If it’s Bounce. Finance Certified, net would be $245,500.
- Development – Unity NFT Game $40,000; HTML5 compiled game $5000; LAMP NFT Plugin/Integration Development: $10,000; Solidity Development Debt for launch: $10,000: Website UX/UI, server, registrations, fees debt: $8000; Marketing Launch Debt: $12000; OpenZeppelin Audit Debt: Unknown, but likely expensive
- Marketing, 15% or – $36757
- Charity — 7% or $17,200
- Treasury for expenses -$106,100 minus OpenZeppelin audit fees
- Signed NFTs
- Merchandise

- Dev: Cross-Chain Yield Aggregation
- Liquidity Pool Faucet (tokens minted, users match with BNB/BUSD directly to timelocked LP provision)

Update

The Grogu team has provided some clarifications.
This project is self-funded, an important distinction going forward, we've absorbed all development and marketing costs and this will not be reimbursed from sale funds. On mainnet launch it'll be "fair use, zero dev issuance", 100% of the presale will go to LP formation which will be permanently locked via multisig. That means we cannot migrate to PCSv3/4/5/whatever, as well as our own dex, without community "multisig" approval. If the community wants the LP to stay in PCSv2 forever, it will. Post-launch we will update you to verify this plan was executed.

Domain Info

Domain Name	NameCheap, Inc.
Registry Domain ID	0d1ef2070ca747f1afe67908911fea4c-DONUTS
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com
Updated Date	2021-07-21 03:46:32 UTC
Creation Date	2021-07-16 03:46:30 UTC
Registry Expiry Date	2022-07-16 03:46:30 UTC
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created one month before the creation of the audit. It will expire in one year.

There is no public billing information, the creator is protected by the privacy settings.

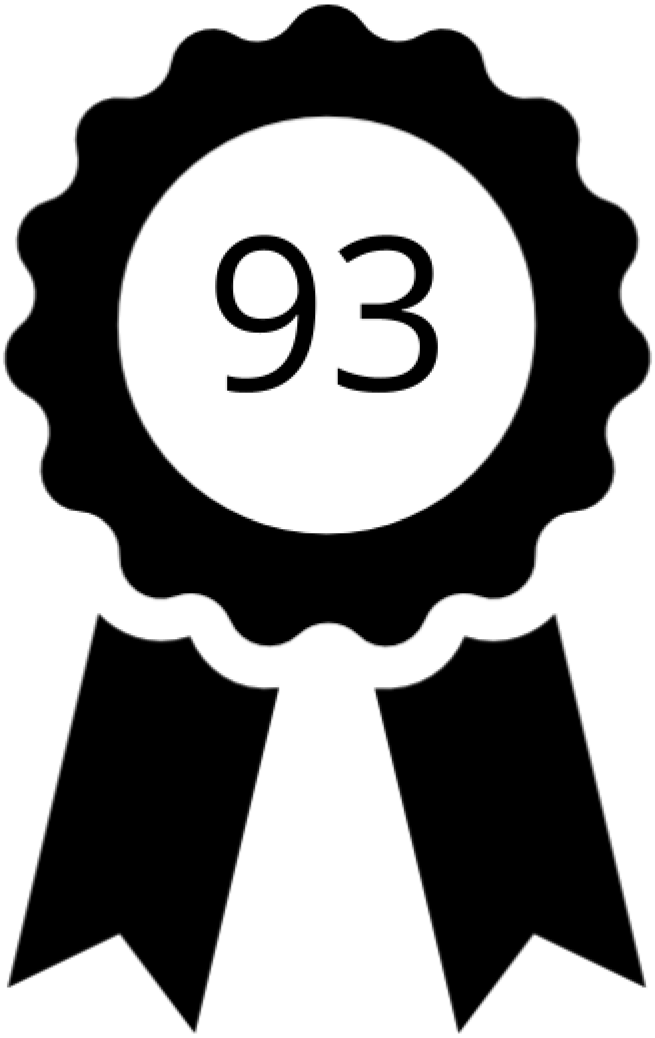
Analysis

Domain	Score	Max
Website Score	19	20
Roadmap Score	4	5
Team Score	8	10
Contract Score	62	65

Coinscope Award system

- 0-49 score points award you with Bronze Badge - high risk.
- 50-89 score points award you with Silver Badge - medium/low risk.
- 90-100 score points award you with Gold Badge - low risk.

Award



Grogu is a low risk project, with a friendly community that grows. There is potential for huge success if they follow their plans. Different network chains and donations may dramatically increase their popularity. Grogu introduces a novel way for choosing the administrators in a voting-based pattern. There is a lot of potential and room for improvements around this mechanism.

Disclaimer

All the content provided in this document is for general information only and should not be used as a financial advice or a reason to buy any investment.

Coinscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

Coinscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Coinscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

Coinscope team disclaims any liability for the resulting losses.

Thank you

The Coinscope.co team,

https://www.coinscope.co/