SOLIDProof
Bring trust into your projects

Blockchain Security | Smart Contract Audits | KYC

MADE IN GERMANY

Audit

Security Assessment
03. December, 2021

For

CHAIN COLOSSEUM
BATTLE OF CHAOS

Disclaimer

SolidProof.io reports are not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team. SolidProof.io do not cover testing or auditing the integration with external contract or services (such as Unicrypt, Uniswap, PancakeSwap etc’...)

SolidProof.io Audits do not provide any warranty or guarantee regarding the absolute bug- free nature of the technology analyzed, nor do they provide any indication of the technology proprietors. SolidProof Audits should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.

SolidProof.io Reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology. Blockchain technology and cryptographic assets present a high level of ongoing risk. SolidProof’s position is that each company and individual are responsible for their own due diligence and continuous security. SolidProof in no way claims any guarantee of security or functionality of the technology we agree to analyze.

Version	Date	Description
1.0	30. November 2021 - 03. December 2021	• Layout project • Automated- /Manual-Security Testing • Summary

Network
Binance Smart Chain (BEP20)

Website
https://chaincolosseum.io/

Telegram
https://t.me/chaincolosseum
https://t.me/ChainColosseumChat

Twitter
https://twitter.com/ChainColosseum_

Medium
https://medium.com/@chaincolosseum

Description

ChainColosseum is a story book like game, and a place where rare items can be traded just like the video games we all played as children. A mystical place where brave men and dragons and demons, fight to keep their existence! Please come join us and explore a dream like nostalgic world!
We're excited to bring ChainColosseum to the world!
We invite you to join our wonderful community!

Project Engagement

During the 24th of November 2021, **ChainColosseum Team** engaged Solidproof.io to audit smart contracts that they created. The engagement was technical in nature and focused on identifying security flaws in the design and implementation of the contracts. They provided Solidproof.io with access to their code repository and whitepaper.

Logo

CHAINCOLOSSEUM
BATTLE OF CHAOS

Contract Link

v1.0
TBA

Vulnerability & Risk Level

Risk represents the probability that a certain source-threat will exploit vulnerability, and the impact of that event on the organization or system. Risk Level is computed based on CVSS version 3.0.

Level	Value	Vulnerability	Risk (Required Action)
Critical	9 - 10	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Immediate action to reduce risk level.
High	7 – 8.9	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.	Implementation of corrective actions as soon aspossible.
Medium	4 – 6.9	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.	Implementation of corrective actions in a certain period.
Low	2 – 3.9	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
Informational	0 – 1.9	A vulnerability that have informational character but is not effecting any of the code.	An observation that does not determine a level of risk

Auditing Strategy and Techniques Applied

Throughout the review process, care was taken to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices. To do so, reviewed line-by-line by our team of expert pentesters and smart contract developers, documenting any issues as there were discovered.

Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:
    i) Review of the specifications, sources, and instructions provided to SolidProof to make sure we understand the size, scope, and functionality of the smart contract.
    ii) Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
    iii) Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to SolidProof describe.

2. Testing and automated analysis that includes the following:
    i) Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
    ii) Symbolic execution, which is analysing a program to determine what inputs causes each part of a program to execute.

3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.

4. Specific, itemized, actionable recommendations to help you take steps to secure your smart contracts.

Used Code from other Frameworks/Smart Contracts (direct imports)

Imported packages:

Dependency / Import Path	Count
@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol	10
@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol	1
@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol	2
@openzeppelin/contracts-upgradeable/proxy/Initializable.sol	12
@openzeppelin/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol	3
@openzeppelin/contracts-upgradeable/token/ERC721/IERC721ReceiverUpgradeable.sol	1
@openzeppelin/contracts-upgradeable/utils/PausableUpgradeable.sol	1
@openzeppelin/contracts-upgradeable/utils/ReentrancyGuardUpgradeable.sol	1
@openzeppelin/contracts/access/Ownable.sol	2
@openzeppelin/contracts/introspection/ERC165Checker.sol	1
@openzeppelin/contracts/math/Math.sol	1
@openzeppelin/contracts/math/SafeMath.sol	10
@openzeppelin/contracts/math/SignedSafeMath.sol	2
@openzeppelin/contracts/token/ERC20/ERC20.sol	2
@openzeppelin/contracts/token/ERC20/ERC20Burnable.sol	2
@openzeppelin/contracts/token/ERC20/IERC20.sol	3
@openzeppelin/contracts/token/ERC20/SafeERC20.sol	4
@openzeppelin/contracts/token/ERC721/IERC721.sol	1
@openzeppelin/contracts/utils/EnumerableSet.sol	1

Tested Contract Files

This audit covered the following files listed below with a SHA-1 Hash.

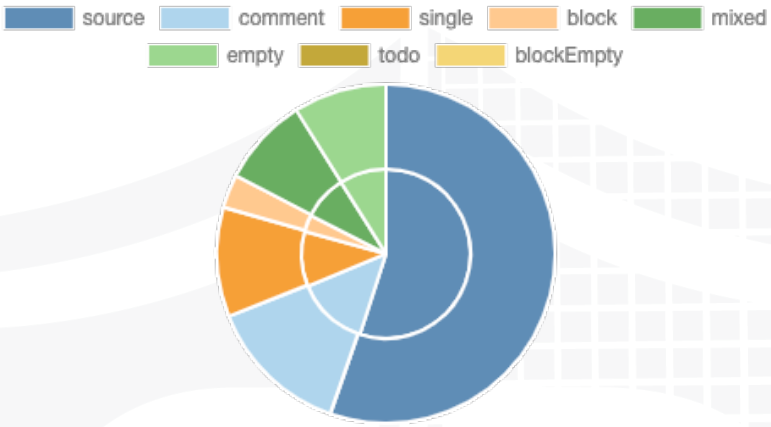
A file with a different Hash has been modified, intentionally or otherwise, after the security review. A different Hash could be (but not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of this review.

v1.0

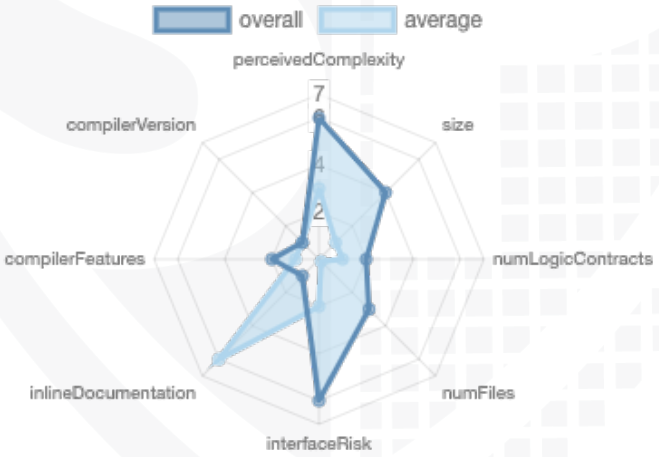
File Name	SHA-1 Hash
contracts/NFTMarket.sol	d27f4e672e9e45ac0cba2c62fbba6170c10ae6c0
contracts/ItemMintTickets.sol	1dfd5f5ac346ff8e2b04c826f652c1d308951da8
contracts/Raid.sol	349eaae4035c265e321780701442a769562e9340
contracts/LPStakingBenefitsUpgradeable.sol	dd0da13a18c1c970a7d1b211276cf4a96b72d0ca
contracts/Bosses.sol	a63a4c34b5bc036b5af1e9307f161845d62bb0d7
contracts/Items.sol	a978260d965088aec2d7942cbd554d94009aa46a
contracts/ColosToken.sol	4bd7e1bb6b7452182efa6cb919f0f95e5c79fd6c
contracts/MasterColosseum.sol	4ed5d0a1a9c06a21c8ec785e8a3a45a87745a742
contracts/interfaces/IRandoms.sol	55c62efdf6f1808e617fe2ed5fa77a387414ca8f
contracts/SkillToken.sol	89bc94b3d0a70984f20807fdeaaca95eea7a3a04
contracts/util.sol	791b907119f2333d59e94063ca8fabe0a0701ff7
contracts/Fight.sol	5b569af161c04ee33cf4690a528e31bb53296723
contracts/interfaces/IPriceOracle.sol	f5768d6ecd85a1a52bae0fc9f8a2bd0f6122b995
contracts/staking/FailsafeUpgradeable.sol	2ddb289fa6a8cf0407f29a4a9efc212b0a085767
contracts/BossMintTickets.sol	1c23a2ea9be7f9bf7f261c9584a15f8cf60ae1c9
contracts/staking/StakingBenefitsUpgradeable.sol	0c2426946ec8ca14052a8250151b892d67acb06a
contracts/BasicPriceOracle.sol	5bb8a49d9dfe124d79e8e4de32d76d6a27b29c5f
contracts/presale/Presale.sol	2249ce2a3c7a0883969681d71c568215a55f81a7
contracts/staking/interfaces/IStakingBenefits.sol	ba5f77a58d2a7c1c90b088858a4501459a36b708
contracts/Tickets.sol	328d11949dfc8d0cfc840e91587416024243b994
contracts/Characters.sol	1239aa3ff697403c5e95a171fb1fc3f05e13367e
contracts/Randoms.sol	de402f9307fd6ef18dc8f89082b3dc3813d2a3d4

Metrics

Source Lines
v1.0



Risk Level
v1.0



Capabilities

Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	17	2	3	1

Exposed Functions

This section lists functions that are explicitly declared public or payable. Please note that getter methods for public stateVars are not included.

Version	Public	Payable
1.0	263	1

Version	External	Internal	Private	Pure	View
1.0	53	245	7	47	137

State Variables

Version	Total	Public
1.0	121	86

Capabilities

Version	Solidity Versions observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts

1.0	`^0.6.0`
	`^0.6.5`
	`^0.6.2`
	`^0.6.12`
	`>=0.4.24`
			`yes`	****
				(0 asm blocks)	

Version	Transfers ETH	Low-Level Calls	DelegateCall	Uses Hash Functions	ECRecover	New/Create/Create2
1.0	`yes`			`yes`		

Scope of Work

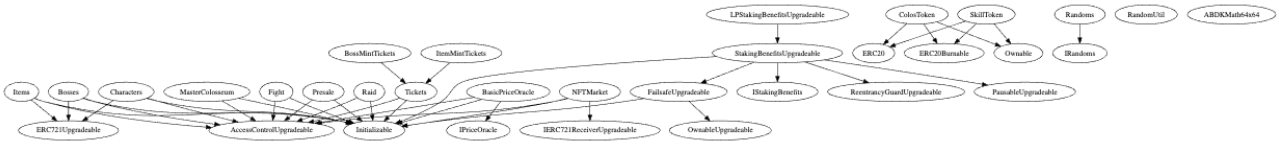
The above token Team provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract (usual the same name as team appended with .sol).

We will verify the following claims:

1. Correct implementation of Token standard
2. Deployer cannot mint any new tokens
3. Deployer cannot burn or lock user funds
4. Deployer cannot pause the contract
5. Overall checkup (Smart Contract Security)

Inheritance Graph

v1.0



Verify Claims

Correct implementation of Token standard

Tested	Verified
✓	✓

Function	Description	Exist	Tested	Verified
TotalSupply	provides information about the total token supply	✓	✓	✓
BalanceOf	provides account balance of the owner's account	✓	✓	✓
Transfer	executes transfers of a specified number of tokens to a specified address	✓	✓	✓
TransferFrom	executes transfers of a specified number of tokens from a specified address	✓	✓	✓
Approve	allow a spender to withdraw a set number of tokens from a specified account	✓	✓	✓
Allowance	returns a set number of tokens from a spender to the owner	✓	✓	✓

Write functions of contract

Bosses

- addTotalGain
- approve
- grantRole
- incrementLose...
- incrementWin...
- initialize
- mint
- renounceRole
- revokeRole
- safeTransferFr...
- safeTransferFr...
- setApprovalFo...
- setATK
- setDEF
- setHP
- setImageUrl
- setInFight
- setLastFightTi...
- setLUK
- setMaxHP
- setMaxLoseCo...
- setName
- setSPD
- transferFrom

BossMintTickets

- giveTicket
- giveTicketByAdmin
- grantRole
- initialize
- initialize
- mintBoss
- renounceRole
- revokeRole
- takeTicketByAdmin
- toggleTicketCanUse

BasicPriceOracle

- grantRole
- initialize
- renounceRole
- revokeRole
- setCurrentPrice

ColosToken

- approve
- burn
- burnFrom
- decreaseAllow...
- increaseAllow...
- mint
- renounceOwn...
- setupGame
- setupSkill
- swapToSkill
- transfer
- transferFrom
- transferOwner...

SkillToken

- approve
- burn
- burnFrom
- decreaseAllow...
- increaseAllow...
- mint
- renounceOwn...
- setSwapPenalty
- setupColos
- setupGame
- setupSale
- swapToColos
- transfer
- transferFrom
- transferOwner...

Characters

- advancedJob
- approve
- boost
- grantRole
- incrementWinCountB...
- incrementWinCountFi...
- incrementWinCountS...
- incrementWinCountT...
- initialize
- migrate_staking
- mint
- renounceRole
- revokeRole
- safeTransferFrom
- safeTransferFrom
- setApprovalForAll
- setFightRound
- setLastFightBossTime...
- setLastFightTimestamp
- setup_cooldown
- transferFrom

Fight

- fight
- grantRole
- initialize
- renounceRole
- revokeRole

Items

- approve
- burn
- giveBossMintTicket
- giveItemMintTicket
- grantRole
- initialize
- migrate_ticket
- mint
- mintItemWithStars
- mintN
- performMintItem
- purchaseBossMintTick...
- purchaseItemMintTicket
- renounceRole
- revokeRole
- safeTransferFrom
- safeTransferFrom
- setApprovalForAll
- setBossMintTicketPrice
- setItemMintTicketPrice
- transferFrom

ItemsMintTickets

- giveTicket
- giveTicketByAdmin
- grantRole
- initialize
- initialize
- mintItem
- mintItemN
- renounceRole
- revokeRole
- takeTicketByAdmin
- toggleTicketCanUse

Presale

- buyTokens
- claimRefTokens
- claimTickets
- claimTokens
- grantRole
- initialize
- renounceRole
- revokeRole
- setBuyerBalance
- setHardCap
- setLimitPerAccount
- setRate
- setRefBalance
- setRefRate
- setReleaseTime
- setTicketBalance
- setTokensPerMaxBuy
- withdraw
- withdrawAll
- withdrawToken

NFTMarket

- addListing
- allowToken
- banUser
- banUsers
- cancelListing
- changeListingPrice
- disallowToken
- grantRole
- initialize
- onERC721Received
- purchaseListing
- renounceRole
- revokeRole
- setDefaultTax
- setDefaultTaxAsPercent
- setTaxOnTokenType
- setTaxOnTokenTypeA...

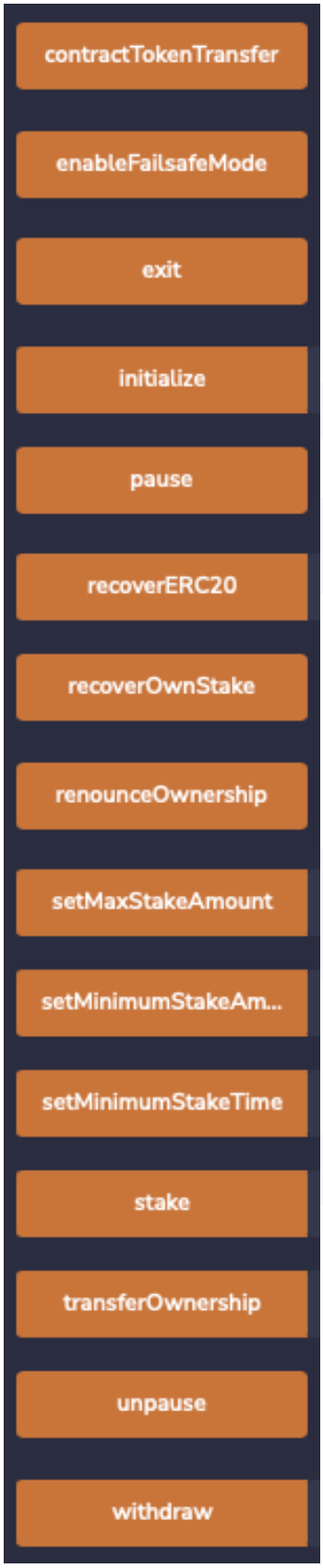
Raid

- fightBoss
- grantRole
- initialize
- renounceRole
- revokeRole
- setFightBossRewardC...
- setFightBossRewardG...
- setFightBossRewardS...
- startRaid
- startRaid1
- startRaid2
- startRaid3
- startRaidId

Random

- setSeed

StakingBenefitsUpgradeable



OnlyOwner functions

Initializer

- BasicPriceOracle
  - constructor
- BossMintTickets
  - constructor
- Bosses
  - initialize
- Characters
  - initialize
- Fight
  - initialize
- ItemMintTickets
  - initialize
- Items
  - initialize
- NFTMarket
  - initialize
- Presale
  - initialize
- Raid
  - initialize
- StakingBenefitsUpgradeable
  - initialize

Restricted

- Bosses

```
mint
setName
setImageUrl
setMaxHP
setHP
setATK
setDEF
setSPD
setLUK
setLastFightTimestamp
setInFight
setMaxLoseCount
addTotalGain
incrementWinCount
incrementLoseCount
```

- Characters

```
mint
setLastFightTimestamp
setLastFightBossTimestamp
setFightRound
incrementWinCountFirst
incrementWinCountSecond
incrementWinCountThird
incrementWinCountBoss
boost
advancedJob
```

- Items

```
mint
mintN
mintItemWithStars
performMintItem
burn
setBossMintTicketPrice
```

```
giveBossMintTicket
setItemMintTicketPrice
giveItemMintTicket
```

- NFTMarket
  - setDefaultTax
  - setTaxOnTokenType
  - setTaxOnTokenTypeAsPercent
  - allowToken
  - disallowToken
  - banUser
  - banUsers
- Presale
  - withdrawAll
  - withdraw
  - withdrawToken
  - setReleaseTime
  - setRate
  - setRefRate
  - setBuyerBalance
  - setRefBalance
  - setTicketBalance
  - setHardCap
  - setLimitPerAccount
  - setTokensPerMaxBuy
- Raid

```
startRaid
startRaid1
startRaid2
startRaid3
startRaidId
```

```
setFightBossRewardSkillBaselineValue
setFightBossRewardColosBaselineValue
setFightBossRewardGasOffsetValue
```

OnlyOwner

- ColorsToken
  - setupSkill
  - setupGame
- SkillToken

setupGame
setupSale
setupColos
setSwapPenalty

- StakingBenefitsUpgradeable
  - RecoverERC20
  - setMinimumStakeAmount
  - setMinimumStakeTime
  - setMaxStakeAmount
  - enableFailsafeMode
  - contractTokenTransfer
  - pause
  - Unpause

OnlyOwnerOrGameOrSkill

- ColorsToken
  - Mint

OnlyOwnerOrGameOrColosOrSale

- SkillToken
  - Mint

FightModifierChecks

- Fight
  - Fight
- Raid
  - fightBoss

TokenAllowed

- NFTMarket
  - addListing

IsValidERC721

- NFTMarket
  - addListing
  - setTaxOnTokenType
  - setTaxOnTokenTypeAsPercent
  - allowToken

isNotListed

- NFTMarket
  - addListing

notBanned

- NFTMarket
  - changeListingPrice
  - cancelListing
  - purchaseListing

isListed

- NFTMarket
  - changeListingPrice
  - cancelListing
  - purchaseListing

isSeller

- NFTMarket
  - changeListingPrice

isSellerOrAdmin

- NFTMarket
  - cancelListing

OnlyNonContract

- Presale
  - buyTokens
  - claimTokens
  - claimRefTokens
  - claimTickets

NormalMode

- StakingBenefitsUpgradeable
  - Stake
  - Withdraw
  - Exit
  - setMinimumStakeAmount
  - setMinimumStakeTime
  - setMaxStakeAmount
  - enableFailsafeMode

NonReentrant

- StakingBenefitsUpgradeable
  - Stake
  - Withdraw

FailsafeMode

- StakingBenefitsUpgradeable
  - RecoverOwnStake

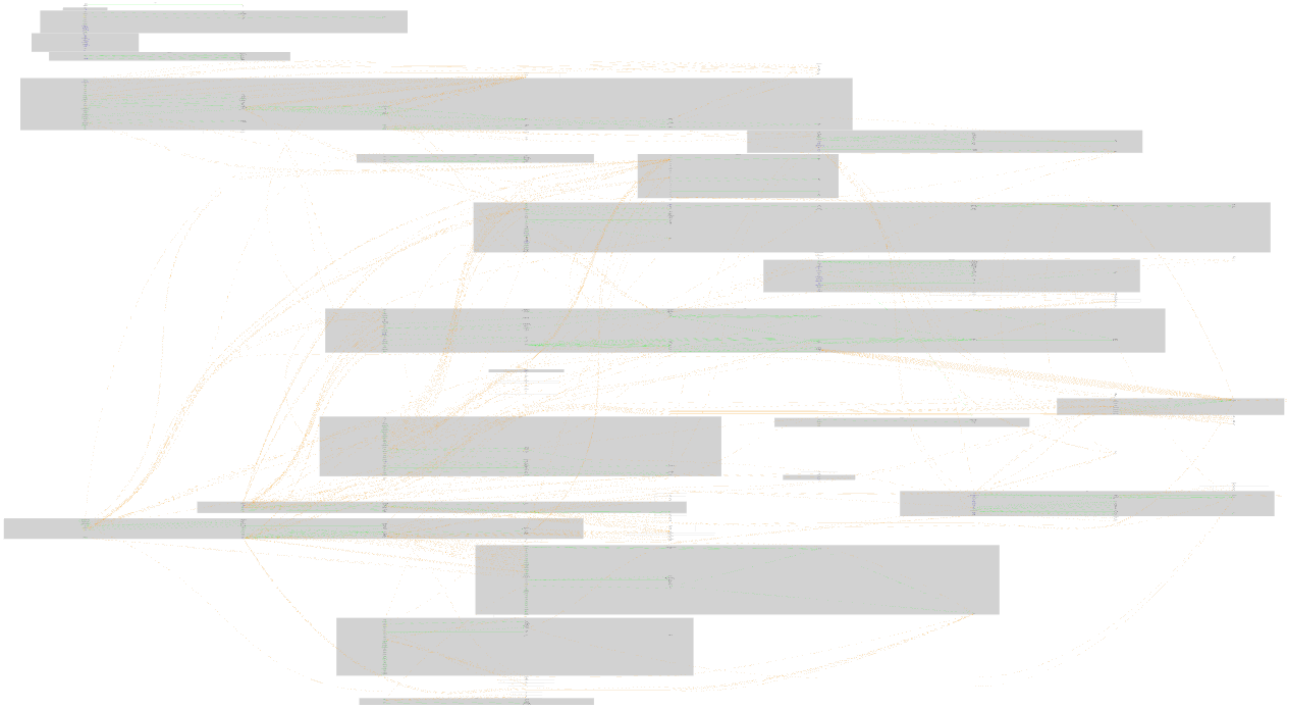
whenPaused

- StakingBenefitsUpgradeable
  - Stake
  - pause
  - Unpause

Comments

- restricted
  - Game admin can set everything from characters without any limitations
- Admin can set cool down in characters without any limitations in setup_cooldown function
- Game admin can set boss variables without limitations

CallGraph



Source Units in Scope
v1.0

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
📝	contracts/NFTMarket.sol	1	—	459	459	384	—	481	🧱
📝	contracts/ItemMintTickets.sol	1	—	32	32	25	—	32	🧱
📝	contracts/Raid.sol	1	—	378	377	315	21	332	🧱
📝	contracts/LPStakingBenefitsUpgradeable.sol	1	—	6	6	4	—	3	—
📝	contracts/Bosses.sol	1	—	238	238	189	1	163	🧱
📝	contracts/Items.sol	1	—	1093	1086	994	597	691	🧱
📝	contracts/ColosToken.sol	1	—	108	108	64	24	58	—
📝	contracts/MasterColosseum.sol	1	—	306	299	228	28	217	🧱
🔍	contracts/interfaces/IRandoms.sol	—	1	6	5	3	1	3	—
📝	contracts/SkillToken.sol	1	—	165	165	124	17	102	—
📚	contracts/util.sol	1	—	73	73	59	1	58	🧱
📝	contracts/Fight.sol	1	—	337	330	270	31	273	🧱
🔍	contracts/interfaces/IPriceOracle.sol	—	1	8	5	3	1	5	—
🎨	contracts/staking/FailsafeUpgradeable.sol	1	—	40	40	32	—	16	—
📚	contracts/abdk-libraries-solidity/ABDKMath64x64.sol	1	—	700	700	424	224	236	—
📝	contracts/BossMintTickets.sol	1	—	33	33	26	—	27	—
📝	contracts/staking/StakingBenefitsUpgradeable.sol	1	—	174	164	137	—	124	—
📝	contracts/BasicPriceOracle.sol	1	—	36	36	28	1	30	🧱
📝	contracts/presale/Presale.sol	1	—	292	284	213	26	219	💰💠🧱
🔍	contracts/staking/interfaces/IStakingBenefits.sol	—	1	18	4	3	—	19	—
📝	contracts/Tickets.sol	1	—	76	76	56	—	42	🧱
📝	contracts/Characters.sol	1	—	439	431	360	15	327	🧱
📝	contracts/Randoms.sol	1	—	16	16	11	1	11	🧱
📝📚🔍🎨	**Totals**	**20**	**3**	**5033**	**4967**	**3952**	**989**	**3469**	💰💠🧱

Legend

Attribute	Description
Lines	total lines of the source unit
nLines	normalized lines of the source unit (e.g. normalizes functions spanning multiple lines)
nSLOC	normalized source lines of code (only source-code lines; no comments, no blank lines)
Comment Lines	lines containing single or block comments
Complexity Score	a custom complexity score derived from code statements that are known to introduce code complexity (branches, loops, calls, external interfaces, ...)

Audit Results

AUDIT PASSED

Critical issues

- no critical issues found -

High issues

- no high issues found -

Medium issues

Issue	File	Type	Line	Description
#1	Fight	Weak PRNG	247, 252	There are services to generate random numbers in Smart contracts like Chainlink VRF (For further information read the doc: https://docs.chain.link/docs/chainlink-vrf/) Do not use `block.timestamp`, `now` or `blockhash` as a source of randomness
#2	Raid	Weak PRNG	176	There are services to generate random numbers in Smart contracts like Chainlink VRF (For further information read the doc: https://docs.chain.link/docs/chainlink-vrf/) Do not use `block.timestamp`, `now` or `blockhash` as a source of randomness

Low issues

Issue	File	Type	Line	Description
#1	All	A floating pragma is set	-	The current pragma Solidity directive starts with ^ or >=.
#2	ColosToken	Missing Zero Address Validation (missing-zero-check)	54	Check that the address is not zero
#3	SkillToken	Missing Zero Address Validation (missing-zero-check)	55, 59	Check that the address is not zero
#4	Fight	State variable visibility is not set	30	It is best practice to set the visibility of state variables explicitly
#5	ItemMintTickets	State variable visibility is not set	8	It is best practice to set the visibility of state variables explicitly
#6	Items	State variable visibility is not set	46	It is best practice to set the visibility of state variables explicitly
#7	MasterColosseum	State variable visibility is not set	41	It is best practice to set the visibility of state variables explicitly
#8	Raid	State variable visibility is not set	37	It is best practice to set the visibility of state variables explicitly
#9	BossMintTickets	State variable visibility is not set	10	It is best practice to set the visibility of state variables explicitly
#10	Raid	Usage of equality comparison instead of assignment	125	Using of comparison instead of assignment Use found = true instead of found == true
#11	Bosses	Missing Events Arithmetic	184	Emit an event for critical parameter changes
#12	Characters	Missing Events Arithmetic	40	Emit an event for critical parameter changes
#13	MasterColosseum	Missing Events Arithmetic	273, 252, 256	Emit an event for critical parameter changes
#14	NFTMarket	Missing Events Arithmetic	417, 420	Emit an event for critical parameter changes

#15	Presale	Missing Events Arithmetic	54, 257	Emit an event for critical parameter changes
#16	Raid	Missing Events Arithmetic	364, 368, 360	Emit an event for critical parameter changes
#17	Tickets	Raw math arithmetic used	49, 66, 71, 54	Use SafeMath library from openzeppelin instead of raw math arithmec if you are using pragma version lower than 0.8.x

Informational issues

Issue	File	Type	Line	Description
#1	StakingBenefitsUpgradeable	State variables that could be declared constant (constable-states)	26	Add the `constant` attributes to state variables that never change
#2	ExperimentBnbBusdLpToken	SPDX license is missing	-	Consider adding a comment containing SPDX-License-Identifier: UNLICENSED
#3	ExperimentBnbToken	SPDX license is missing	-	Consider adding a comment containing SPDX-License-Identifier: UNLICENSED
#4	ExperimentBusdToken	SPDX license is missing	-	Consider adding a comment containing SPDX-License-Identifier: UNLICENSED
#5	ExperimentToken	SPDX license is missing	-	Consider adding a comment containing SPDX-License-Identifier: UNLICENSED
#6	ItemMintTickets	SPDX license is missing	-	Consider adding a comment containing SPDX-License-Identifier: UNLICENSED
#7	Items	SPDX license is missing	-	Consider adding a comment containing SPDX-License-Identifier: UNLICENSED

#8	LPStakingBenefitsUpgradeable	SPDX license is missing	-	Consider adding a comment containing SPDX-License-Identifier: UNLICENSED
#9	Raid	SPDX license is missing	-	Consider adding a comment containing SPDX-License-Identifier: UNLICENSED
#10	Tickets	SPDX license is missing	-	Consider adding a comment containing SPDX-License-Identifier: UNLICENSED
#11	StakingBenefitsUpgradeable	SPDX license is missing	-	Consider adding a comment containing SPDX-License-Identifier: UNLICENSED
#12	FailsafeUpgradeable	SPDX license is missing	-	Consider adding a comment containing SPDX-License-Identifier: UNLICENSED
#13	LPStakingBenefitsUpgradeable	Empty code block	-	Empty code block
#14	Bosses	Functions that are not used	58	Remove unused functions
#15	Items	Functions that are not used	84	Remove unused functions
#16	Presale	Functions that are not used	104	Remove unused functions
#17	Util	Functions that are not used	26, 35, 50, 40, 45, 18	Remove unused functions
#18	Items	Unused state variables	47	Remove unused state variables

Commented Code exist

There are some instances of code being commented out in the following files that should be removed:

File	Line	Comment
ColorsToken	63	// holdersInfo[_from].avgTransactionBlock = _getAvgTransactionBlock(_from, holdersInfo[_from], _amount, true);

ColorsToken	74-79	// uint256 penalty = getPenaltyPercent(_holderAddress);
		// if(penalty == 0){
		// return _colosAmount;
		// }
		// return _colosAmount.sub(_colosAmount.mul(penalty).div(1e12));

Fight	270	// base = atk / 2 - def / 4
Fight	272	// int256 damage = int256(atk.div(2));
Fight	278	// range = base / 16 + 2
MasterColosseum	174-177	// function advancedCharacterJob(uint256 char) public isCharacterOwner(char) oncePerBlock(msg.sender) requestPayFromPlayer(advancedCharacterJobFee) { // _payContract(msg.sender, advancedCharacterJobFee); // characters.advancedJob(char, msg.sender); // }

MasterColosseum	184-198	// function mintItemN() public onlyNonContract oncePerBlock(msg.sender) requestPayFromPlayer(mintItemNFee) {
		// _payContract(msg.sender, mintItemNFee);
		// items.mintN(msg.sender, 11, uint256(keccak256(abi.encodePacked(blockhash(block.number - 1), msg.sender))));
		// }
		
		// function burnItemN(uint256[] memory itemIds) public isItemsOwner(itemIds) oncePerBlock(msg.sender) {
		// require(itemIds.length > 0);
		// uint256 totalStars = 0;
		// for (uint i = 0; i < itemIds.length; i++){
		// totalStars = totalStars + items.getStars(itemIds[i]) + 1;
		// items.burn(itemIds[i]);
		// }
		// uint256 num = totalStars.div(10);
		// items.giveItemMintTicket(msg.sender, num);
		// }

Presale	31	// mapping(address => mapping(address => Bought)) boughtByRef; // referral address : bought by referral
Presale	105	// require(useToken.balanceOf(msg.sender) >= useAmount, "insufficient funds.");

Recommendation

Remove the commented code, or address them properly.

Audit Comments

03. December 2021:

- ABDK-librariy was not provided to solidproof
  - We had to add it manually from a library (https://github.com/abdk-consulting/abdk-libraries-solidity)
- ColosToken
  - In line 64 and line 65
    - _amount and skillAmount are the same because the _swapSkillAmount function in line 72 has no effect for the _colosAmount, this will burn _amount and mint the same amount in skillToken

```
72  function _swapSkillAmount(address _holderAddress, uint256 _colosAmount) internal view returns (uint256 expectedSkill){
73      require(balanceOf(_holderAddress) >= _colosAmount, "Not enough COLOS");
74      // uint256 penalty = getPenaltyPercent(_holderAddress);
75      // if(penalty == 0){
76      //     return _colosAmount;
77      // }
78
79      // return _colosAmount.sub(_colosAmount.mul(penalty).div(1e12));
80      return _colosAmount;
81  }
```

    - Line 104 has no effect because amount is not used after line 104
- Read whole report for more information

SWC Attacks

ID	Title	Relationships	Status
SWC-136	Unencrypted Private Data On-Chain	CWE-767: Access to Critical Private Variable via Public Method	PASSED
SWC-135	Code With No Effects	CWE-1164: Irrelevant Code	NOT PASSED
SWC-134	Message call with hardcoded gas amount	CWE-655: Improper Initialization	PASSED
SWC-133	Hash Collisions With Multiple Variable Length Arguments	CWE-294: Authentication Bypass by Capture-replay	PASSED
SWC-132	Unexpected Ether balance	CWE-667: Improper Locking	PASSED
SWC-131	Presence of unused variables	CWE-1164: Irrelevant Code	PASSED
SWC-130	Right-To-Left-Override control character (U+202E)	CWE-451: User Interface (UI) Misrepresentation of Critical Information	PASSED
SWC-129	Typographical Error	CWE-480: Use of Incorrect Operator	PASSED
SWC-128	DoS With Block Gas Limit	CWE-400: Uncontrolled Resource Consumption	PASSED

SWC-127	Arbitrary Jump with Function Type Variable	CWE-695: Use of Low-Level Functionality	PASSED
SWC-125	Incorrect Inheritance Order	CWE-696: Incorrect Behavior Order	PASSED
SWC-124	Write to Arbitrary Storage Location	CWE-123: Write-what-where Condition	PASSED
SWC-123	Requirement Violation	CWE-573: Improper Following of Specification by Caller	PASSED
SWC-122	Lack of Proper Signature Verification	CWE-345: Insufficient Verification of Data Authenticity	PASSED
SWC-121	Missing Protection against Signature Replay Attacks	CWE-347: Improper Verification of Cryptographic Signature	PASSED
SWC-120	Weak Sources of Randomness from Chain Attributes	CWE-330: Use of Insufficiently Random Values	PASSED
SWC-119	Shadowing State Variables	CWE-710: Improper Adherence to Coding Standards	PASSED
SWC-118	Incorrect Constructor Name	CWE-665: Improper Initialization	PASSED
SWC-117	Signature Malleability	CWE-347: Improper Verification of Cryptographic Signature	PASSED

SWC-116	Timestamp Dependence	CWE-829: Inclusion of Functionality from Untrusted Control Sphere	PASSED
SWC-115	Authorization through tx.origin	CWE-477: Use of Obsolete Function	PASSED
SWC-114	Transaction Order Dependence	CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')	PASSED
SWC-113	DoS with Failed Call	CWE-703: Improper Check or Handling of Exceptional Conditions	PASSED
SWC-112	Delegatecall to Untrusted Callee	CWE-829: Inclusion of Functionality from Untrusted Control Sphere	PASSED
SWC-111	Use of Deprecated Solidity Functions	CWE-477: Use of Obsolete Function	PASSED
SWC-110	Assert Violation	CWE-670: Always-Incorrect Control Flow Implementation	PASSED
SWC-109	Uninitialized Storage Pointer	CWE-824: Access of Uninitialized Pointer	PASSED
SWC-108	State Variable Default Visibility	CWE-710: Improper Adherence to Coding Standards	NOT PASSED
SWC-107	Reentrancy	CWE-841: Improper Enforcement of Behavioral Workflow	PASSED
SWC-106	Unprotected SELFDESTRUCT Instruction	CWE-284: Improper Access Control	PASSED

SWC-105	Unprotected Ether Withdrawal	CWE-284: Improper Access Control	PASSED
SWC-104	Unchecked Call Return Value	CWE-252: Unchecked Return Value	PASSED
SWC-103	Floating Pragma	CWE-664: Improper Control of a Resource Through its Lifetime	NOT PASSED
SWC-102	Outdated Compiler Version	CWE-937: Using Components with Known Vulnerabilities	PASSED
SWC-101	Integer Overflow and Underflow	CWE-682: Incorrect Calculation	PASSED
SWC-100	Function Default Visibility	CWE-710: Improper Adherence to Coding Standards	PASSED

Solid Proofed

Blockchain Security | Smart Contract Audits | KYC

MADE IN GERMANY