Audit Report

Cryptit

January 2022

Github https://github.com/CryptITAustria/ElysiumClub

Audited by © coinscope

Table of Contents

Contract Review

The contract audit will review the files that are included in the github repository

Github	https://github.com/CryptITAustria/ElysiumClub
Commit	780236bc6b8c64329881f9d371d5574666c17238

The audit review is based on the following files

File	**SHA256**
Playmate Final.sol	757b4e5986d32a501ec367fc1571ea2c4e390f73808a30a8eb7664d8732d502f

Audit Updates

Initial Audit	26th January 2022
Corrected	

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L09	Dead Code Elimination
	L11	Unnecessary Boolean equality
	L07	Missing Events Arithmetic

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L697,L688,L636 and 2 more

Description

Public functions that are never called by the contract should be declared external to save gas.

```
tokenByIndex
tokenOfOwnerByIndex
transferOwnership
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L774

Description

Constant state variables should be declared constant to save gas.

```
maxMints
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L788,L787,L1086 and 4 more

Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
_raribleBeneficiary
_beneficiary
_maxUserMints
...
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L601,L588,L236 and 22 more

Description

Functions that are not used in the contract, and make the code's size bigger.

```
toHexString
sub
mod
...
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L1067

Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
require(bool,string)(isCollabPartner[collabAddress] == true,Not marked as
partner)
```

Recommendation

Remove the equality to the boolean constant.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1086,L940

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
maxUserMints = _maxUserMints
_collabPrice = price
```

Recommendation

Emit an event for critical parameter changes.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
LibPart	Library			
	hash	Internal		
RoyaltiesV2	Interface			
	getRaribleV2Royalties	External		-
AbstractRoyalties	Implementation			
	_saveRoyalties	Internal	✓	
	_updateAccount	Internal	✓	
	_onRoyaltiesSet	Internal	✓	
IERC165	Interface			
	supportsInterface	External		-
ERC165	Implementation	IERC165		
	supportsInterface	Public		-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	

	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	_verifyCallResult	Private		
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
IERC721	Interface	IERC165		
	balanceOf	External		-
	ownerOf	External		-
	safeTransferFrom	External	✓	-
	transferFrom	External	✓	-
	approve	External	✓	-
	getApproved	External		-
	setApprovalForAll	External	✓	-
	isApprovedForAll	External		-
	safeTransferFrom	External	✓	-
IERC721Enumerable	Interface	IERC721		
	totalSupply	External		-
	tokenOfOwnerByIndex	External		-
	tokenByIndex	External		-

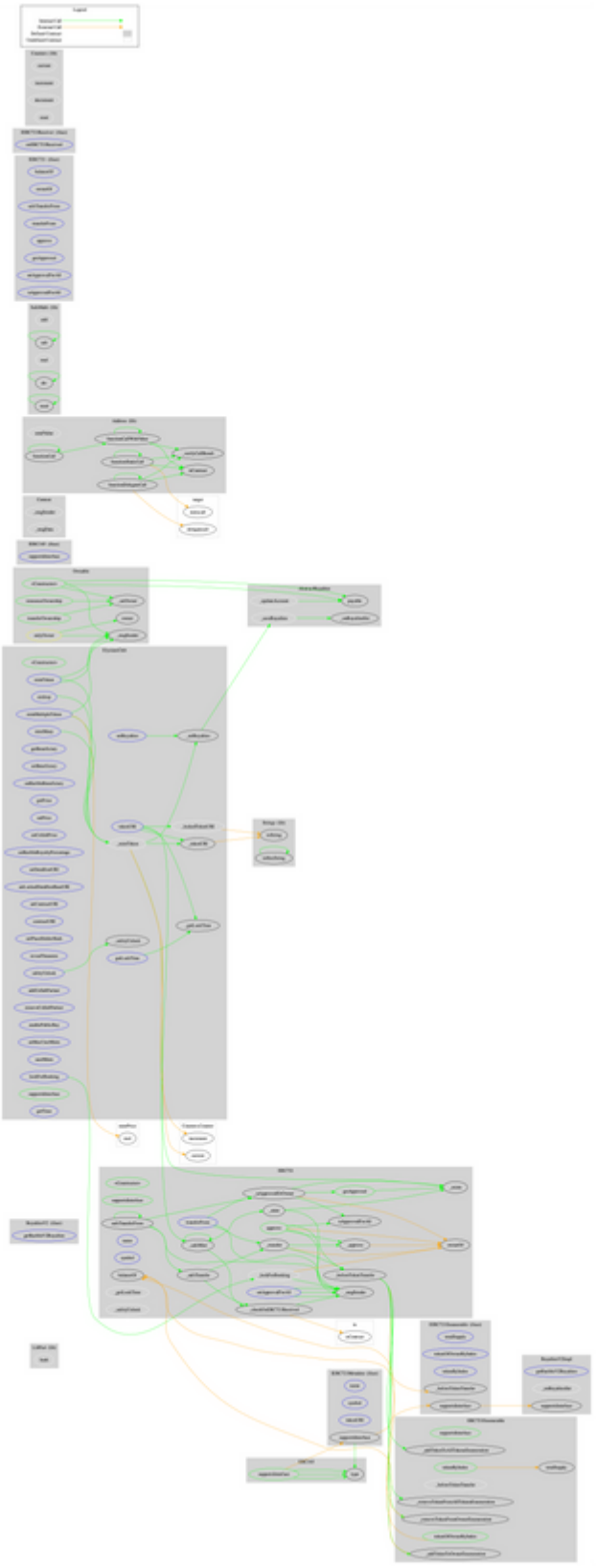
IERC721Metadata	Interface	IERC721		
	name	External		-
	symbol	External		-
	tokenURI	External		-
IERC721Receiver	Interface			
	onERC721Received	External	✓	-
ERC721	Implementation	Context, ERC165, IERC721, IERC721Metadata		
	<Constructor>	Public	✓	-
	supportsInterface	Public		-
	balanceOf	Public		-
	ownerOf	Public		-
	name	External		-
	symbol	External		-
	getApproved	Public		-
	setApprovalForAll	External	✓	-
	isApprovedForAll	Public		-
	approve	Public	✓	-
	transferFrom	External	✓	-
	safeTransferFrom	External	✓	-
	safeTransferFrom	Public	✓	-
	_safeTransfer	Internal	✓	
	_getLockTime	Internal		
	_lockForBooking	Internal	✓	
	_safetyUnlock	Internal	✓	
	_exists	Internal		
	_isApprovedOrOwner	Internal		
	_safeMint	Internal	✓	
	_safeMint	Internal	✓	
	_mint	Internal	✓	

	_transfer	Internal	✓	
	_approve	Internal	✓	
	_checkOnERC721Received	Private	✓	
	_beforeTokenTransfer	Internal	✓	
LibRoyaltiesV2	Library			
RoyaltiesV2Impl	Implementation	AbstractRoyalties, RoyaltiesV2		
	getRaribleV2Royalties	External		-
	_onRoyaltiesSet	Internal	✓	
Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	
Counters	Library			
	current	Internal		
	increment	Internal	✓	
	decrement	Internal	✓	
	reset	Internal	✓	
ERC721Enumerable	Implementation	ERC721, IERC721Enumerable		
	supportsInterface	Public		-
	tokenOfOwnerByIndex	Public		-
	totalSupply	Public		-

	tokenByIndex	Public		-
	_beforeTokenTransfer	Internal	✓	
	_addTokenToOwnerEnumeration	Private	✓	
	_addTokenToAllTokensEnumeration	Private	✓	
	_removeTokenFromOwnerEnumeration	Private	✓	
	_removeTokenFromAllTokensEnumeration	Private	✓	
ElysiumClub	Implementation	ERC721Enumerable, Ownable, RoyaltiesV2Impl		
	<Constructor>	Public	✓	ERC721 Ownable
	_mintToken	Internal	✓	
	mintToken	External	Payable	-
	mintMultipleToken	External	Payable	-
	airdrop	External	✓	onlyOwner
	mintMany	External	✓	onlyOwner
	getBeneficiary	External		-
	setBeneficiary	External	✓	onlyOwner
	setRaribleBeneficiary	External	✓	onlyOwner
	getPrice	External		-
	setPrice	External	✓	onlyOwner
	setCollabPrice	External	✓	onlyOwner
	setRaribleRoyaltyPercentage	External	✓	onlyOwner
	setDataHostURI	External	✓	onlyOwner
	setLockedDataHostBaseURI	External	✓	onlyOwner
	setContractURI	External	✓	onlyOwner
	contractURI	External		-
	_tokenURI	Internal		
	_lockedTokenURI	Internal		
	tokenURI	External		-
	_setRoyalties	Internal	✓	
	setRoyalties	External	✓	onlyOwner
	setPlaceHolderHash	External	✓	onlyOwner

	revealTreasures	External	✓	onlyOwner
	safetyUnlock	External	✓	onlyOwner
	addCollabPartner	External	✓	onlyOwner
	removeCollabPartner	External	✓	onlyOwner
	enablePublicBuy	External	✓	onlyOwner
	setMaxUserMints	External	✓	onlyOwner
	userMints	External		-
	getLockTime	External		-
	lockForBooking	External	✓	-
	supportsInterface	Public		-
	getTime	External		-

Contract Flow



Summary

CryptIt contract implements the fundamental NFT functionality enriched with some additional features.

The contract owner may define the “collaborative partners”. The collaborative partners is a list of addresses that will be charged differently in the mint functionality.

The users have the ability to lock their holdings for a period of time. During this period the locked asset cannot be transferred.

Users have the ability to mint one or multiple tokens. The contract owner has the ability to mint multiple tokens to a specific address or many tokens to many addresses.

Disclaimer

All the content provided in this document is for general information only and should not be used as 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.

The 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.

The Coinscope team disclaims any liability for the resulting losses.

About Coinscope

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Coinscope is aiming to make crypto discoverable and efficient globally. It provides all the essential tools to assist users draw their own conclusions.

The Coinscope.co team

https://www.coinscope.co