Security Audit by Aleksander Wójcik (aleksanderw1992)

During crystalize.dev bootcamp I was asked to perform peer review using slither security tool for security. I found only minor problems - no severe violations. For audit I used both automated tool and my own experience and knowledge

Issues:

Issue 1:

Severity: Optimization
Confidence: High

Slither output:

```
 ArcadeGamesNFT.mintItem() (contracts/ArcadeGamesNFT.sol#48-50)
withdraw() should be declared external:
        - ArcadeGamesNFT.withdraw() (contracts/ArcadeGamesNFT.sol#70-80)
setBaseURI(string) should be declared external:
        - ArcadeGamesNFT.setBaseURI(string)
(contracts/ArcadeGamesNFT.sol#86-88)
```

Code:

```
56    function withdraw()
57        public
58        onlyOwner
59    {
60        payable(msg.sender).transfer(address(this).balance);
61    }
```

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

Recommendation
Use the external attribute for functions never called from the contract.

Reference:

https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external

Issue 2:

Severity: Informational
Confidence: High

Code:

```
2 pragma solidity ^0.8.13;
```

Contract uses Solidity version ^0.8.13 and libraries (OpenZeppelin) '^0.8.0', '^0.8.1'. Try to unify the Solidity version.
Description
Detect whether different Solidity versions are used.

Recommendation
Use one Solidity version.

Reference:
https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Issue 3:

Severity: Informational
Confidence: Medium

Slither Output:

```
ArcadeGamesNFT.mintItem(uint256) (contracts/ArcadeGamesNFT.sol#56-65) has
costly operations inside a loop:
        - tokenIdCounter ++ (contracts/ArcadeGamesNFT.sol#62)
ERC721Enumerable._removeTokenFromAllTokensEnumeration(uint256)
(node_modules/@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumera
ble.sol#144-162) has costly operations inside a loop:
```

Code:

```
45        for(uint256 i=0; i < _amount; i++){
46            tokenIdCounter++; // references to state variable
47            _safeMint(to, tokenIdCounter);
48            _setTokenURI(tokenIdCounter, URI);
49        }
```

Description
Costly operations inside a loop might waste gas, so optimizations are justified.

Reference:
https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Logs:

ethsec@de836f1f26a5:/home/training$ slither .

'npx hardhat compile --force' running
Compiled 14 Solidity files successfully

Solidity 0.8.13 is not fully supported yet. You can still use Hardhat, but some features, like stack traces, might not work correctly.

Learn more at https://hardhat.org/hardhat-runner/docs/reference/solidity-support

ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#394-416) ignores return value by IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,data) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#401-412)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#394-416) has external calls inside a loop: IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,data) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#401-412)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop

Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#401)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#394-416) potentially used before declaration: retval == IERC721Receiver.onERC721Received.selector (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#402)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#403)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#394-416) potentially used before declaration: reason.length == 0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#404)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#403)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#394-416) potentially used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#409)

Reference:
https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

ERC721._checkOnERC721Received(address,address,uint256,bytes)
(node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#394-416) uses assembly
- INLINE ASM
(node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#408-410)
Address.verifyCallResult(bool,bytes,string)
(node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
- INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity are used:
- Version used: ['^0.8.0', '^0.8.1', '^0.8.13']
- ^0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Receiver.sol#4)
- ^0.8.0
(node_modules/@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol#4)
- ^0.8.0
(node_modules/@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol#4)
- ^0.8.0
(node_modules/@openzeppelin/contracts/token/ERC721/extensions/IERC721Enumerable.sol#4)
- ^0.8.0
(node_modules/@openzeppelin/contracts/token/ERC721/extensions/IERC721Metadata.sol#4)
- ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Strings.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/ERC165.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4)
- ^0.8.13 (contracts/ArcadeGamesNFT.sol#2)
Reference:
https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

ArcadeGamesNFT.mintItem(uint256) (contracts/ArcadeGamesNFT.sol#56-65) has costly operations inside a loop:
- tokenIdCounter ++ (contracts/ArcadeGamesNFT.sol#62)

ERC721Enumerable._removeTokenFromAllTokensEnumeration(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol#144-162) has costly operations inside a loop:
	- delete _allTokensIndex[tokenId] (node_modules/@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol#160)
ERC721Enumerable._removeTokenFromAllTokensEnumeration(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol#144-162) has costly operations inside a loop:
	- _allTokens.pop() (node_modules/@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol#161)
ERC721Enumerable._removeTokenFromOwnerEnumeration(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol#119-137) has costly operations inside a loop:
	- delete _ownedTokensIndex[tokenId] (node_modules/@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol#135)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

ArcadeGamesNFT._burn(uint256) (contracts/ArcadeGamesNFT.sol#111-116) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Receiver.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/extensions/IERC721Enumerable.sol#4) allows old versions

Pragma version^0.8.0
(node_modules/@openzeppelin/contracts/token/ERC721/extensions/IERC721Metadata.sol#4)
allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old
versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old
versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Strings.sol#4) allows old
versions
Pragma version^0.8.0
(node_modules/@openzeppelin/contracts/utils/introspection/ERC165.sol#4) allows old versions
Pragma version^0.8.0
(node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4) allows old versions
Pragma version^0.8.13 (contracts/ArcadeGamesNFT.sol#2) necessitates a version too recent to
be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.13 is not recommended for deployment
Reference:
https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256)
(node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
	- (success) = recipient.call{value: amount}()
(node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string)
(node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
	- (success,returndata) = target.call{value: value}(data)
(node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string)
(node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
	- (success,returndata) = target.staticcall(data)
(node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string)
(node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
	- (success,returndata) = target.delegatecall(data)
(node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Low level call in ArcadeGamesNFT.withdraw() (contracts/ArcadeGamesNFT.sol#70-80):
	- (sent) = address(msg.sender).call{value: address(this).balance}()
(contracts/ArcadeGamesNFT.sol#78)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter ArcadeGamesNFT.mintItem(uint256)._amount (contracts/ArcadeGamesNFT.sol#56)
is not in mixedCase
Parameter ArcadeGamesNFT.setBaseURI(string)._baseTokenURI
(contracts/ArcadeGamesNFT.sol#86) is not in mixedCase

Reference:
https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

renounceOwnership() should be declared external:
- Ownable.renounceOwnership()
(node_modules/@openzeppelin/contracts/access/Ownable.sol#61-63)
transferOwnership(address) should be declared external:
- Ownable.transferOwnership(address)
(node_modules/@openzeppelin/contracts/access/Ownable.sol#69-72)
name() should be declared external:
- ERC721.name()
(node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#79-81)
symbol() should be declared external:
- ERC721.symbol()
(node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#86-88)
approve(address,uint256) should be declared external:
- ERC721.approve(address,uint256)
(node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#112-122)
setApprovalForAll(address,bool) should be declared external:
- ERC721.setApprovalForAll(address,bool)
(node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#136-138)
transferFrom(address,address,uint256) should be declared external:
- ERC721.transferFrom(address,address,uint256)
(node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#150-159)
safeTransferFrom(address,address,uint256) should be declared external:
- ERC721.safeTransferFrom(address,address,uint256)
(node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#164-170)
tokenOfOwnerByIndex(address,uint256) should be declared external:
- ERC721Enumerable.tokenOfOwnerByIndex(address,uint256)
(node_modules/@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol#37-40)
tokenByIndex(uint256) should be declared external:
- ERC721Enumerable.tokenByIndex(uint256)
(node_modules/@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol#52-55)
mintItem() should be declared external:
- ArcadeGamesNFT.mintItem() (contracts/ArcadeGamesNFT.sol#48-50)
withdraw() should be declared external:
- ArcadeGamesNFT.withdraw() (contracts/ArcadeGamesNFT.sol#70-80)
setBaseURI(string) should be declared external:
- ArcadeGamesNFT.setBaseURI(string) (contracts/ArcadeGamesNFT.sol#86-88)

Reference:
https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
. analyzed (14 contracts with 78 detectors), 48 result(s) found