Quantstamp Security Assessment Certificate

QUANTSTAMP VERIFIED SECURITY CERTIFICATE

September 20th 2022 — Quantstamp Verified

Sandbox Staking Contracts

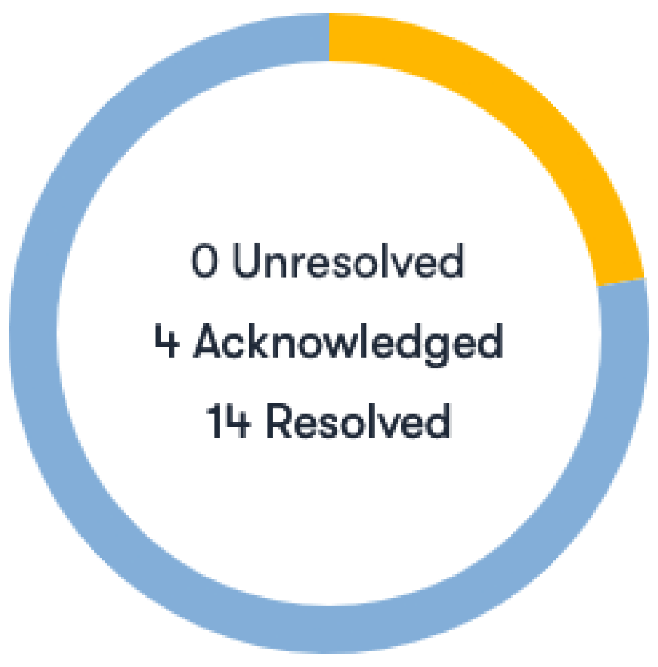
This audit report was prepared by Quantstamp, the leader in blockchain security.

Executive Summary

Type	Decentralized Gaming Platform
Auditors	Alejandro Padilla Gaeta, Research Engineer Souhail Mssassi, Research Engineer Rabib Islam, Research Engineer
Timeline	2022-05-20 through 2022-09-19
EVM	Arrow Glacier
Languages	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	ERC20RewardPool Documentation
Documentation Quality	Medium
Test Quality	Medium
Source Code	

Repository	Commit
sandbox-smart-contracts	b5b1a5c
sandbox-smart-contracts	f7c1bcc
sandbox-smart-contracts	8071735
sandbox-smart-contracts	7a078f3
sandbox-smart-contracts	80bb838
sandbox-smart-contracts	aa156cf

Total Issues	18	(14 Resolved)
High Risk Issues	1	(1 Resolved)
Medium Risk Issues	5	(5 Resolved)
Low Risk Issues	7	(6 Resolved)
Informational Risk Issues	2	(0 Resolved)
Undetermined Risk Issues	3	(2 Resolved)



High Risk	The issue puts a large number of users' sensitive information at risk, or is reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
Medium Risk	The issue puts a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or is reasonably likely to lead to moderate financial impact.
Low Risk	The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low-impact in view of the client's business circumstances.
Informational	The issue does not post an immediate risk, but is relevant to security best practices or Defence in Depth.
Undetermined	The impact of the issue is uncertain.

Unresolved	Acknowledged the existence of the risk, and decided to accept it without engaging in special efforts to control it.
Acknowledged	The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings).
Fixed	Adjusted program implementation, requirements or constraints to eliminate the risk.
Mitigated	Implemented actions to minimize the impact or likelihood of the risk.

Summary of Findings

This report presents the results of the audit that Quantstamp performed on the Sandbox staking contracts. It's important to highlight that the Sandbox codebase is quite big and the contracts reviewed represent only a small part of the overall functionality; all code outside the Sandbox staking contracts is considered out-of-scope for this audit.

The audit identified **18 issues** across all levels of severity, including 1 of high- and 5 of medium-severity. The Sandbox team provided good documentation explaining how the staking contracts are supposed to work. However, the documentation across the contracts audited is inconsistent. For tests, the coverage for most files is acceptable, but there are a few contracts like `ERC2771HandlerV2.sol` that lack sufficient coverage.

Re-audit update:
As noted by the Sandbox team, the issue "Admin calls can be called as meta transactions" was a false positive. Therefore, it was removed from the report. The re-audit also **uncovered 3 more issues**, while **2 of the original issues were marked as mitigated** as they were only partially fixed (`QSP-3` and `QSP-7`). We recommend addressing all of them.
Note: Given one issue was removed and others were added, the order of the issues has changed.

Re-audit update (2):
After discussing with the Sandbox team, the issue "Not clear who is funding the rewards" was marked as a false positive and removed from the report. The rest of the issues have been fixed or acknowledged. Only `QSP-3` remains partially fixed (there's one branch that could still lead to Denial-of-Service). We believe the risk is low, but the fix is quite simple so we recommend addressing it before deploying the code into production.

Re-audit update (3):
The latest commit fixed `QSP-3`. All issues identified by the audit have been either fixed or acknowledged.

Fixes review (4):
The Sandbox team released a new commit (`80bb838`) that introduced two fixes:

- The getters in the `LockRules` contract (`getRemainingTimelockClaim`, `getRemainingTimelockClaim`, and `getRemainingTimelockDeposit`) used to wrongly return the **time since** the last timelock. Now they have been fixed to return the **time left** in the timelock.
- Before this commit, the contribution of a user would be calculated based on the NFT/assets that the user owned when the `stake` method was called. Because of this he would continue to accrue rewards even if he did not own any NFT/assets anymore. To address this, the Sandbox team introduced a new business rule to calculate the contribution considering the NFT/assets that the user holds at that moment (`_computeContribution` now calls the `maxStakeAllowedCalculator` method).

The Quantstamp team reviewed the commit and only found a discrepancy between the way the max stake is calculated in the `checkRequirements` modifier and the `maxStakeAllowedCalculator` method. This issue was fixed on commit `aa156cf`.

ID	Description	Severity	Status
QSP-1	Admin can steal staked funds	High	Fixed
QSP-2	Contract cannot be paused	Medium	Fixed
QSP-3	Unbounded Iteration Exposes Denial-of-Service	Medium	Fixed
QSP-4	Faulty checks in `RequirementRules`	Medium	Fixed
QSP-5	Possible to set values outside reasonable limits	Medium	Fixed
QSP-6	Wrong calculation of the ERC1155 stake	Medium	Fixed
QSP-7	Possible to renounce ownership	Low	Fixed
QSP-8	Contract ownership might not be transferred to rightful owner	Low	Acknowledged
QSP-9	Validation missing throughout the codebase	Low	Fixed
QSP-10	`ERC2771Handler` not checking `msg.data` length	Low	Fixed
QSP-11	Changing tokens without checking balances	Low	Fixed
QSP-12	Unlocked Pragma	Low	Fixed
QSP-13	Get methods in `LockRules` contract can revert	Low	Fixed
QSP-14	Staked funds can be frozen	Informational	Acknowledged
QSP-15	Use of `block.timestamp`	Informational	Acknowledged
QSP-16	Confusing Privileged Roles and Ownership	Undetermined	Fixed
QSP-17	NFT Double-counting	Undetermined	Fixed
QSP-18	Contributions might not be up to date	Undetermined	Acknowledged

Quantstamp Audit Breakdown

Quantstamp's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.

DISCLAIMER:
This audit focused only on the following files: ERC20RewardPool.sol, IContributionRules.sol, ContributionRules.sol, LockRules.sol, RequirementsRules.sol, ERC2771HandlerV2.sol, StakeTokenWrapper.sol, IContributionCalculator.sol, IRewardCalculator.sol, and SafeMathWithRequire.sol

Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow / underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Methodology

The Quantstamp auditing process follows a routine series of steps:

1. Code review that includes the following
   i. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
   ii. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
   iii. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Quantstamp describe.
2. Testing and automated analysis that includes the following:
   i. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
   ii. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

Toolset

The notes below outline the setup and steps performed in the process of this audit.

Setup

Tool Setup:

- Slither v0.8.2

Steps taken to run the tools:

1. Install the Slither tool: `pip3 install slither-analyzer`
2. Run Slither from the project directory: `slither .`

Notes: 1) We only ran Slither on the `src/solc_0.8/*` folder. 2) We only reviewed issues reported for the files under the scope of this audit.

Findings

QSP-1 Admin can steal staked funds

Severity: *High Risk*

Status: Fixed

File(s) affected: `ERC20RewardPool.sol`

Description: The `recoverFunds` method allows the admin to transfer the `_rewardToken` balance to any address of his choosing. If the `_stakeToken` happens to be the same as the `_rewardToken`, this could allow the admin to effectively "steal" the stakers' staked funds (along with all the rewards).

Recommendation: If the `_stakeToken` is the same as the `_rewardToken`, subtract `_totalSupply` in the calculation of funds to be recovered. Also, consider adding additional checks to ensure that the admin can only call this method during an emergency and not arbitrarily.

Update: The Sandbox team added checks to ensure that when the `stakeToken` is the same as the `rewardToken` only the balance corresponding to rewards can be recovered. Additionally, the `recoverFunds` method can only be called when the contract is paused (note that an admin can always pause the contract).

QSP-2 Contract cannot be paused

Severity: *Medium Risk*

Status: Fixed

File(s) affected: `ERC20RewardPool.sol`

Description: Several methods in the `ERC20RewardPool` contract are annotated with the `whenNotPaused` modifier. However, the `_paused` flag is never changed (directly or indirectly) in the codebase. This means that in case of an emergency the owner won't be able to pause the contract. **Note:** Previous versions of the `Pausable` contract in the OpenZeppelin library used to contain public `pause` and `unpause` methods, but now they are internal.

Recommendation: Consider adding external methods that only an owner (or admin) can call to pause/unpause the contract.

Update: Methods `pause` and `unpause` have been added to the `ERC20RewardPool` contract. These methods allow the owner to pause the contract or return it to normal.

QSP-3 Unbounded Iteration Exposes Denial-of-Service

Severity: *Medium Risk*

Status: Fixed

File(s) affected: `ContributionRules.sol`, `RequirementsRules.sol`, `ERC20RewardPool.sol`

Description: Before executing the `stake` method, the contract calls the `checkRequirements` modifier to validate that the staking account meets the necessary requirements. This modifier in turn calls the `checkERC1155MinStake`, `checkERC721MinStake`, `getERC721MaxStake`, and `getERC1155MaxStake` methods, each of which calculates different balances of the account by iterating over lists of contracts and IDs. Unfortunately, none of these iterations have upper bounds for the number of contracts or number of IDs. Therefore, the `stake` function may become unable to run due to gas limits being exceeded during these checks.
The same issue occurs in `computeMultiplier` when it calls `multiplierBalanceOfERC721` and `multiplierBalanceOfERC1155`.

Recommendation: Add upper bounds to how many contracts can have rules and how many IDs are allowed to be included in those rules. Additionally, add end-user and technical documentation explaining this risk.

Update: The Sandbox team introduced limits to bound the iteration of items in the `ContributionRules` methods. However, no limits have been added to the methods in the `RequirementsRules` contract. Therefore, the risk around Denial-of-Service remains. Also, no technical documentation has been added to explain the risk.

Update 2: The Sandbox team introduced limits to bound the iteration of items in the `RequirementRules` methods. This reduces considerably the risk of Denial-of-Service (DoS). However, there still is a code branch that can bypass these limits. If the `balanceOf` flag is set to true when calling the `setERC721RequirementList` method, it is possible to set a list of `ids` of any size (the `idsLimit` is never checked on `L119` when `balanceOf` is set to true). If the list of `ids` is too large, the calls to `getERC721BalanceId` on `L334` could run out-of-gas. Given `setERC721RequirementList` can only be called by the owner and that there are other checks in the codebase, the risk of DoS is low. However, the fix is quite easy (add a check on `L119`) so we recommend addressing the issue before deploying the code to production.

Update 3: The latest commit introduced a check to the `setERC721RequirementList` method to ensure that the `ids` list is empty when the `balanceOf` flag is set to `true`. This addresses the risk of Denial-of-Service. However, it also means that when checking the `ERC721` stake of an account (via the `checkAndGetERC721Stake` method), each rule will only consider `balanceOf` or `ids`, but not both. Make sure this is acceptable according to your business rules.

QSP-4 Faulty checks in `RequirementRules`

Severity: *Medium Risk*

Status: Fixed

File(s) affected: `RequirementsRules.sol`

Description: The `RequirementsRules` contract has two methods (`isERC721MemberRequirementList` and `isERC1155MemberRequirementList`) that verify if there is a `RequirementRule` registered for a given token address. However, both of these methods will always return false due to the following faulty checks:

- `isERC721MemberRequirementList:L215` - `return (_listERC721Index.length == 0) && (...);`
- `isERC1155MemberRequirementList:L219` - `return (_listERC1155Index.length == 0) && (...);`

Recommendation: Change the condition in both lines (`L215` and `L219`) from `==` to `!=`.

Update: The conditions on both faulty checks have been fixed. We recommend adding additional test cases to ensure that these checks always work as expected.

QSP-5 Possible to set values outside reasonable limits

Severity: *Medium Risk*

Status: Fixed

File(s) affected: `LockRules.sol`, `RequirementRules.sol`, `ContributionRules.sol`

Description: There are several methods across the codebase that allow an owner (or admin) to set important values that are never validated. This makes it possible to provide values (by mistake or maliciously) that could put the operation of the contracts at risk. This list contains the most relevant cases (might not be an exhaustive list):

- `LockRules.sol` : `lockPeriodInSecs` (for `timeLockClaim`, `lockDeposit`, and `lockWithdraw`) - It is possible to set the different time locks to absurdly low or high values; a very small value will allow the lock to expire immediately, while a very large value would allow funds to be held practically "forever".
- `LockRules.sol`: `amountLockClaim.amount` - If the amount is set to a very high value it might be possible to hold the values forever.
- `ContributionRules`: multipliers (`multiplierLimitERC271` and `multiplierLimitERC1155`) - If these limits are set to 0, only the `amountStaked` will be taken into account. Otherwise, if the value is too high the amount computed could end up too high (above 100% of the amount staked).

Recommendation: Determine reasonable limits (as constants) for each of the values that can be set by admins/owners and validate that the values they set fall within those limits.

Update: Code was refactored to prevent setting values outside reasonable limits. **Note:** It's still possible to set the `multiplierLimitERC721` and the `multiplierLimitERC1155` to 0.

QSP-6 Wrong calculation of the ERC1155 stake

Severity: *Medium Risk*

Status: Fixed

File(s) affected: `RequirementsRules.sol`

Description: While calculating the ERC1155 stake of an account, the `checkAndGetERC1155Stake` wrongly overwrites the `_maxStake` variable on every iteration (`L331`). Hence, the `_maxStake` variable will only have the stake that the given address has on the last contract on the `_listERC1155Index` list, instead of the stake across all contracts.

Recommendation: Refactor `L331` to accumulate the `_maxStake` on every iteration instead of overwriting it. Also, include additional tests to ensure that the stake is calculated properly.

Update: The Sandbox team refactored the code to properly keep track of the ERC1155 `_maxStake`. We still recommend adding additional tests to prevent any future regressions.

QSP-7 Possible to renounce ownership

Severity: *Low Risk*

Status: Fixed

File(s) affected: `ERC20RewardPool.sol`, `ContributionRules.sol`, `LockRules.sol`

Description: The `ERC20RewardPool` contract implements the `AccessControl` and the `Ownable` contracts to only allow privileged users (owner or admin) to execute certain key operations. However, by implementing these contracts, the `ERC20RewardPool` contract is also inheriting the `renounceOwnership` and the `renounceRole` methods. If either the owner or the admin renounced their status by calling one of these methods, it would not be possible to perform owner/admin-specific functionality on the contract anymore.

Recommendation: Override the `renounceRole` and `renounceOwnership` such that they revert when called.

Update: The `renounceOwnership` method has been overwritten where necessary (`ERC20RewardPool.sol` and `ContributionRules.sol`) to revert when called. Therefore, it's not possible to renounce ownership anymore.

QSP-8 Contract ownership might not be transferred to rightful owner

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `RequirementsRules.sol`, `ContributionRules.sol`, `LockRules.sol`, `ERC20RewardPool.sol`

Description: Contracts inheriting from `Ownable` or `OwnableUpgradeable` will have the address that deployed the contract as the owner. If the deployment script does not change that right away, the deployment address will keep ownership (along with all of its privileges) instead of the rightful owner.

Recommendation: If it is not intended for the deployer to be the owner, we recommend transferring ownership of the contracts to the rightful owner directly in the constructor (or initializer function).

Update: The Sandbox team acknowledges the risk, but decided to address it through the deployment script instead of on-chain. It is up to the Sandbox team to ensure that the script always transfers the ownership to the right account.

QSP-9 Validation missing throughout the codebase

Severity: *Low Risk*

Status: Fixed

File(s) affected: `ContributionRules.sol`, `StakeTokenWrapper.sol`, `ERC20RewardPool.sol`

Description: The following list contains the methods across the codebase that do not validate their input parameters (list might not be exhaustive):

- `ERC20RewardPool`
  - . `setTrustedForwarder` (`L103`) does not check that the new `trustedForwarder` is a valid contract. An invalid relayer would prevent the contracts from receiving meta transactions.
  - . Neither the constructor (`L72`) nor the `setRewardToken` (`L92`) check that the `rewardToken` is not an empty address.
  - . The `set*` methods in the contract (`setStakeToken`, `setTrustedForwarder`, `setContributionRules`, `setRewardCalculator`) do not validate that the given addresses are not empty.
- `StakeTokenWrapper`:
  - . Possible to construct a wrapper for an empty IERC20 token address (`L16`).
  - . Possible to stake (`L20`) and withdraw (`L26`) 0 tokens.
- `ContributionRules`:
  - . Unlike the `setERC1155MultiplierList`, the `setERC721MultiplierList` does not validate the given arrays.

Recommendation: While most of these cases are innocuous, we recommend reviewing the list and adding validation where necessary.

Update: The Sandbox team added validation to most places where it was missing.
Notes:

- The `trustedForwarder` variable is still not validated in the `ERC20RewardPool` constructor. However, we consider the impact of this quite low as the owner of the contract can always update the contract to a valid value.
- The new validation in the `ERC20RewardPool` contract prevents the owner from removing the `contributionRules` and `rewardCalculator`. Make sure that is acceptable according to your business rules.

QSP-10 `ERC2771Handler` not checking `msg.data` length

Severity: *Low Risk*

Status: Fixed

File(s) affected: `ERC2771Handler.sol`

Description: The `ERC2771Handler` contract allows a trusted forwarder to send a message on behalf of an end user. When it does so, it will wrap the address of the original sender in the last 20 bytes of the `msg.data`. Therefore, any call from the the `trustedForwarder` has to be at least 24 bytes in length (4 bytes for the ABI function selector + 20 bytes of the original sender address). However, neither the `_msgSender` nor the `_msgData` methods verify that the minimum length is respected; this could lead to unexpected behaviour if data is below 24 bytes.
Note: It is very unlikely for this to ever be a problem as the `ERC2771Handler` contract will only accept calls coming from a trusted forwarder.

Recommendation: Consider validating on both the `_msgSender` and `_msgData` that the `msg.data` has the right length.

Update: The Sandbox team replaced the old `ERC2771Handler.sol` with a new one (`ERC2771HandlerV2.sol`) that checks that all messages received from the `trustedForwarder` are always large enough for meta-transactions.

QSP-11 Changing tokens without checking balances

Severity: *Low Risk*

Status: Fixed

File(s) affected: `ERC20RewardPool.sol`

Description: The `setStakeToken` (`L98`) and `setRewardToken` (`L91`) allow an admin to change the corresponding underlying tokens. However, the methods never check if the pool has enough balance available in the new token to fulfill all the stake and/or reward withdrawals.

Recommendation: Consider adding a check to the `setStakeToken` and `setRewardToken` to validate that the pool has enough balances available in the new token to fulfill the corresponding stake and/or reward withdrawals.

Update: The Sandbox team added checks to ensure that before replacing any token, the new tokens always have enough balance to fulfill the necessary withdrawals.

QSP-12 Unlocked Pragma

Severity: *Low Risk*

Status: Fixed

File(s) affected: `ERC2771Handler.sol`

Related Issue(s): SWC-103

Description: Every Solidity file specifies in the header a version number of the format `pragma solidity (^)0.8.*`. The caret (^) before the version number implies an unlocked pragma, meaning that the compiler will use the specified version *and above*, hence the term "unlocked".

Recommendation: For consistency and to prevent unexpected behavior in the future, we recommend to remove the caret to lock the file onto a specific Solidity version.

Update: The Sandbox team replaced the contract that had an unlocked pragma (`ERC2771Handler.sol`) with another one (`ERC2771HandlerV2.sol`) that has a fixed pragma.

QSP-13 Get methods in `LockRules` contract can revert

Severity: *Low Risk*

Status: Fixed

File(s) affected: `LockRules.sol`

Description: There are several methods in the `LockRules` contract that should return the amount of time remaining on a timelock (`getRemainingTimelockClaim`, `getRemainingTimelockWithdraw`, and `getRemainingTimelockDeposit`). When the timelock has not expired, they will return a negative value, causing the methods to revert (they are all returning a `uint256`).

Recommendation: Refactor the code in these methods to return 0 if the timelock has already expired. Otherwise, return the time remaining.

QSP-14 Staked funds can be frozen

Severity: *Informational*

Status: Acknowledged

File(s) affected: `ERC20RewardPool.sol`

Description: If the `owner` pauses the `ERC20RewardPool` contract, then users who have staked their funds will not be able to withdraw their funds for the duration of the pause. While this is probably the intended behaviour (the contract is likely to only be paused due to an emergency), end users are unlikely to be aware that their funds might be locked longer than expected because of these pauses.

Recommendation: Properly explain in user-facing documentation that users' funds can be frozen in case the contract is paused due to an emergency. Also, mention that it is the `owner` who determines when to pause a contract and how long to keep it in that state.

Update: The Sandbox team will update their end-user documentation to explain that staked tokens might get frozen in case of an emergency (as determined by the owner of the contract).

QSP-15 Use of `block.timestamp`

Severity: *Informational*

Status: Acknowledged

File(s) affected: `ERC20RewardPool.sol`, `LockRules.sol`

Description: The core functionality of the staking contracts relies on `block.timestamp`. While this is common for many smart contracts, it must be noted that miners individually set the timestamp of a block. Therefore, a malicious miner could manipulate the timestamp for his own benefit by up to 900 seconds.

Recommendation: Consider implementing tests to ensure that a timestamp discrepancy won't affect the staking contracts. Otherwise, inform end users (via public-facing documentation) that the timestamp in the system might have an error as high as 900 seconds and warn them to take necessary precautions.

Update: The Sandbox team understands the risk, but decided not to address in code. Instead, they will inform their users about the risk through documentation.

QSP-16 Confusing Privileged Roles and Ownership

Severity: Undetermined

Status: Fixed

File(s) affected: `ContributionRules.sol`, `LockRules.sol`, `RequirementsRules.sol`, `ERC20RewardPool.sol`

Description: Like many other smart contracts, the `ERC20RewardPool` has some functionality that is restricted to addresses with special privileges. However, the setup to control access to the restricted functionality is very confusing; privileged methods in the `ERC20RewardPool` can only be called by an admin (using `AccessControl` methods), while methods in parent contracts can only be called by an owner (using `Ownable` methods). It is unclear if the admin/owner are the same person or if they should have different responsibilities; this could easily lead to errors and misconfiguration.

Recommendation: We strongly recommend choosing to use either the `AccessControl` or `Ownable` contracts to avoid any confusion and misconfiguration. Otherwise, if you consider it's necessary to use both, explicitly document what the responsibilities of the owners/admins are and review the methods in the contract (and contracts it is inheriting) to make sure they are annotated with the right modifier (`hasRole` or `onlyOwner`). Also, we recommend disclosing to end users that the control of privileged parts of the contract is centralized on the owner/admins.

Update: The Sandbox team replaced the checks in the `ERC20RewardPool` contract that were using `AccessControl` with modifiers from the `Ownable` contract.

QSP-17 NFT Double-counting

Severity: Undetermined

Status: Fixed

File(s) affected: `RequirementsRules.sol`, `ContributionRules.sol`

Description: The `getERC721MaxStake` method calculates the max amount that an account can stake by adding up the stake it has in each `ERC721` contract. Internally, the account's stake on each `ERC721` contract is calculated by summing the total number of NFTs in that contract held by the account (`balanceOf`), plus the number of NFTs the account has of a specific subset of IDs in the contract (`balanceOfId`). Thus, the NFTs in the designated subset are being counted twice.
The same thing happens in `multiplierBalanceOfERC721`.

Recommendation: Determine whether this is the intended functionality. If not, consider refactoring the code to prevent the double counting.

Update: The source code was refactored to prevent NFT double counting.

QSP-18 Contributions might not be up to date

Severity: Undetermined

Status: Acknowledged

Description: The staking contracts use the balance of tokens that an account holds to determine if the account meets the requirements to stake and to calculate its contribution (this will be used to determine the rewards). For this to work properly, the staking contracts need to be informed as soon as an account holdings change. Otherwise, the calculations might be wrong (for example, two accounts might be using the same NFT as contribution).

Recommendation: There's a comment in code stating that an external agent must call the `computeContribution` method to update users' contributions. Make sure to identify who the external agent is, how it is chosen, and what its responsibilities are (how fast must it call the update, what happens if it doesn't update it fast enough, etc). If the agent is considered not sufficient to accurately track all changes, make sure to put additional measures in place to limit the impact of inaccurate balances (for example, the `amountStaked` should account for most of the contributions).

Update: The Sandbox team has decided to acknowledge the risk, as they believe that outdated contributions will not have a meaningful impact on the results.

Automated Analyses

Slither

The Slither tool detected 984 results. Most of them were either false positives or were outside the scope of this audit. The remaining results have been incorporated in other sections of the report.

Notes:

- We only ran Slither on the `src/solc_0.8/*` folder.
- We only reviewed issues reported for the files under the scope of this audit.

Code Documentation

1. Documentation is very inconsistent through the codebase. We recommend documenting all code according to the NatSpec Format.
2. The description of the `IRewardCalculator.sol:L5` only mentions the `SandRewardPool`. This has to be updated to mention the `ERC20RewardPool` too.
3. There are several typos:
   - `ERC20RewardPool.sol:L232` - 'stack' should be 'stake'.
   - `LockRules.sol:L34` - 'used' should be 'user'.
   - `ERC20RewardPool.sol:L207` - 'distributes' should be 'distributed'.
4. **(Re-audit)** `ERC20RewardPool.sol - L28`: The comment in this line has either a typo ("giving" instead of "given") or the comment is incomplete.
5. **(Re-audit)** There are two `ERC2771Handler` files that are almost identical (`ERC2771Handler.sol` and `ERC2771HandlerV2.sol`). The only difference is that the v2 file contains additional checks to ensure that `msg.data` is always large enough. Consider adding documentation to explain when to use each file.
6. **(Re-audit)** `ERC20RewardPool.sol`: Consider adding documentation to the `isContractAndAdmin` modifier to highlight that it checks that the given address is a contract and that the caller of the method is the owner of the `ERC20RewardPool`, and not of the given contract.
7. **(Re-audit)** `ERC20RewardPool.sol - L228`: Typo “make” should be “makes”.

Adherence to Best Practices

1. The `ERC2771Handler.sol` is not meant to be used on its own. Therefore, mark it as `abstract`.
2. `ERC20RewardPool.sol` and `ContributionRules.sol` - When dealing with large numbers consider using the `_` to make them more readable. For example, the constant `DECIMALS_18` on `ERC20RewardPool` (`L47`) can be rewritten as `1_000_000_000_000_000_000`.
3. `RequirementsRules.sol` - The `checkRequirements` modifier is calculating the balance of ERC1155 and ERC721 twice. Due to the iterations this can be an expensive operation, so calling it twice might waste too much gas. Consider refactoring the code to calculate those balances only once.
4. The `ContributionRules.sol` contract does not implement the `IContributionRules` interface. Consider implementing it to ensure that there's no discrepancy between the methods defined in the interface and in the contract.
5. `ERC20RewardPool.sol` - Instead of containing the functionality of the `_withdrawRewards` function with an if, use a `require` statement to prevent execution if the reward is 0.
6. `RequirementsRules.sol` - Setting the `balanceOfId` variable on `L226` is unnecessary, as it will always be overwritten on `L233`.
7. Consider updating the `lastWithdraw` on `L295` before `L294`, as that will end up calling external code. While reentrancy is not possible due to all the previous checks, it is always better to update state before any method calling external code.
8. **(Re-audit)** `ERC20RewardPool.sol` - `L409` and `L419`: These lines are using the `1e24` number without any explanation, which makes the code hard to read. Consider moving this value to a constant with an appropriate name.
9. **(Re-audit)** `ContributionRules.sol` (`L16-L18`) and `LockRules.sol` (`L10:L11`) - The variables `idsLimit`, `contractsLimit`, `maxMultiplier`, `timeLockLimit`, and `amountLockLimit` are only set once when the contract is created. Afterward, they are only read from. Therefore, consider setting them as constants.
10. **(Re-audit)** `erc20RewardPool.test.ts` - The file is missing test cases to check that `pause`, `unpause`, and `renounceOwnership` can only be called by the owner.

Test Results

Test Suite Results

We were able to run the test suite successfully and all tests passed without a problem.
Fixes review (4): Tests still passing without any problem.

```
Network Info
============
> HardhatEVM: v2.6.1
> network:    hardhat



  Asset:ERC1155
    bouncerAdmin
Nothing to compile
       ✓ can't set address 0 to bouncerAdmin (4147ms)
    mint
       ✓ minting an item results in a TransferSingle event (129ms)
    transfers
       ✓ transferring one instance of an item results in an ERC1155 TransferSingle event (157ms)
       ✓ transferring multiple instances of an item results in an ERC1155 TransferSingle event (159ms)
       ✓ transferring zero instances of an item results in an ERC1155 TransferSingle event (150ms)
       ✓ transferring an item with 1 supply does not result in an ERC1155 TransferBatch event (149ms)
       ✓ transferring an item with >1 supply does not result in an ERC1155 TransferBatch event (155ms)
       ✓ can be transferred to a normal address (158ms)
       ✓ cannot be transferred to zero address (122ms)
       ✓ cannot transfer more items than you own (116ms)
       ✓ cannot transfer an item with supply 1 that you do not own (111ms)
       ✓ cannot transfer an item that you do not own (115ms)
       ✓ cannot transfer more item of 1 supply (113ms)
       ✓ cannot transfer to a contract that does not accept ERC1155 (114ms)
       ✓ cannot transfer multiple instances of an item to a contract that does not accept ERC1155 (116ms)
       ✓ cannot transfer an item of supply 1 to a contract that does not accept ERC1155 (115ms)
       ✓ cannot transfer an item of supply 1 to a contract that does not return the correct ERC1155_IS_RECEIVER value (116ms)
       ✓ cannot transfer an item of supply >1 to a contract that does not return the correct ERC1155_IS_RECEIVER value (118ms)
    batch transfers
       ✓ transferring an item with 1 supply results in an ERC1155 BatchTransfer event (159ms)
       ✓ transferring an item with >1 supply results in an ERC1155 BatchTransfer event (164ms)
       ✓ transferring zero items with 1 supply results in an ERC1155 BatchTransfer event (151ms)
       ✓ transferring zero items with >1 supply results in an ERC1155 BatchTransfer event (146ms)
       ✓ transferring empty list results in an ERC1155 BatchTransfer event (144ms)
       ✓ transferring multiple items results in an ERC1155 BatchTransfer event (217ms)
       ✓ transferring multiple items including zero amount results in an ERC1155 BatchTransfer event (191ms)
       ✓ transferring an item with 1 supply with batch transfer does not result in a TransferSingle event (165ms)
       ✓ transferring an item with >1 supply with batch transfer does not result in a TransferSingle event (165ms)
       ✓ can use batch transfer to send tokens to a normal address (161ms)
       ✓ cannot batch transfer the same token twice and exceed the amount owned (117ms)
       ✓ can use batch transfer to send token twice if there is sufficient amount owned (181ms)
       ✓ cannot batch transfer tokens to zeroAddress (117ms)
       ✓ cannot batch transfer tokens if array lengths do not match (113ms)
       ✓ cannot batch transfer more than the amount owned (111ms)
       ✓ cannot batch transfer more items of 1 supply (110ms)
       ✓ cannot batch transfer to a contract that does not accept ERC1155 (117ms)
       ✓ cannot batch transfer to a contract that does not return the correct magic value (638ms)
       ✓ can batch transfer to a contract that does accept ERC1155 and which returns the correct magic value (240ms)
       ✓ can batch transfer item with 1 or more supply at the same time (174ms)
       ✓ can obtain balance of batch (379ms)
    approvalForAll
       ✓ setting approval results in ApprovalForAll event (376ms)
       ✓ setting approval fails if sender is operator (123ms)
       ✓ operator cannot transfer without approval (117ms)
       ✓ operator can transfer after approval (187ms)
       ✓ operator cannot transfer after approval is removed (612ms)
    supportsInterface
       ✓ contract claims to supports ERC165 (128ms)
       ✓ contract does not claim to support random interface (118ms)
       ✓ contract does not claim to support invalid interface (112ms)
    ordering
       ✓ transfer empty array (149ms)
       ✓ transfer multiple items in any order (i) (194ms)
       ✓ transfer multiple items in any order (ii) (191ms)
       ✓ transfer multiple items in any order (iii) (182ms)
       ✓ transfer multiple items in any order (iv) (229ms)
       ✓ transfer multiple items in any order (v) (193ms)
       ✓ transfer multiple items in any order (vi) (211ms)
       ✓ transfer multiple items in any order (vii) (193ms)
       ✓ transfer multiple items in any order (viii) (195ms)
       ✓ transfer multiple items in any order twice (i) (253ms)
       ✓ transfer multiple items in any order twice (ii) (308ms)
       ✓ transfer multiple items in any order twice (iii) (265ms)
       ✓ transfer multiple items in any order twice (iv) (305ms)
       ✓ transfer multiple items in any order twice (v) (269ms)
       ✓ transfer multiple items in any order twice (vi) (275ms)

  Asset:ERC721
    non existing NFT
       ✓ transfering a non existing NFT fails (51ms)
       ✓ tx balanceOf a zero owner fails (38ms)
       ✓ call balanceOf a zero owner fails (38ms)
       ✓ tx ownerOf a non existing NFT fails (38ms)
       ✓ call ownerOf a non existing NFT fails (38ms)
       ✓ tx getApproved a non existing NFT fails
       ✓ call getApproved a non existing NFT fails
    balance
       ✓ balance is zero for new user
       ✓ balance return correct value (138ms)
    mint
       ✓ mint result in a transfer from 0 event (44ms)
       ✓ mint for gives correct owner (46ms)
    burnAsset
       ✓ burn result in a transfer to 0 event (77ms)
       ✓ burn result in ownerOf throwing (79ms)
    transfer
       ✓ transfering one NFT results in one erc721 transfer event (67ms)
       ✓ transfering one NFT change to correct owner (71ms)
```

```
      ✓ transfering one NFT increase new owner balance (72ms)
      ✓ transfering one NFT decrease past owner balance (71ms)
      ✓ transfering from without approval should fails
      ✓ transfering to zero address should fails
      ✓ transfering to a contract that do not accept erc721 token should not fail (103ms)
    safeTransfer
      ✓ safe transfering one NFT results in one erc721 transfer event (66ms)
      ✓ safe transfering to zero address should fails (41ms)
      ✓ safe transfering one NFT change to correct owner (69ms)
      ✓ safe transfering from without approval should fails
      ✓ safe transfering to a contract that do not accept erc721 token should fail (49ms)
      ✓ safe transfering to a contract that do not return the correct onERC721Received bytes shoudl fail (47ms)
      ✓ safe transfering to a contract that do not implemented onERC721Received should fail (49ms)
      ✓ safe transfering to a contract that return the correct onERC721Received bytes shoudl succeed (91ms)
    safeTransfer with empty bytes
      ✓ data:0x : safe transfering one NFT results in one erc721 transfer event (72ms)
      ✓ data:0x : safe transfering to zero address should fails
      ✓ data:0x : safe transfering one NFT change to correct owner (70ms)
      ✓ data:0x : safe transfering from without approval should fails (39ms)
      ✓ data:0x : safe transfering to a contract that do not accept erc721 token should fail (47ms)
      ✓ data:0x : safe transfering to a contract that do not return the correct onERC721Received bytes shoudl fail (47ms)
      ✓ data:0x : safe transfering to a contract that do not implemented onERC721Received should fail (46ms)
      ✓ data:0x : safe transfering to a contract that return the correct onERC721Received bytes shoudl succeed (90ms)
    safeTransfer with data
      ✓ data:0xff56fe3422 : safe transfering one NFT results in one erc721 transfer event (67ms)
      ✓ data:0xff56fe3422 : safe transfering to zero address should fails (38ms)
      ✓ data:0xff56fe3422 : safe transfering one NFT change to correct owner (73ms)
      ✓ data:0xff56fe3422 : safe transfering from without approval should fails
      ✓ data:0xff56fe3422 : safe transfering to a contract that do not accept erc721 token should fail (45ms)
      ✓ data:0xff56fe3422 : safe transfering to a contract that do not return the correct onERC721Received bytes shoudl fail (45ms)
      ✓ data:0xff56fe3422 : safe transfering to a contract that do not implemented onERC721Received should fail (47ms)
      ✓ data:0xff56fe3422 : safe transfering to a contract that return the correct onERC721Received bytes shoudl succeed (87ms)
    ERC165
      ✓ claim to support erc165
      ✓ claim to support base erc721 interface
      ✓ claim to support erc721 metadata interface (38ms)
      ✓ does not claim to support random interface
      ✓ does not claim to support the invalid interface
    Approval
      ✓ approving emit Approval event (60ms)
      ✓ removing approval emit Approval event (79ms)
      ✓ approving update the approval status (62ms)
      ✓ cant approve if not owner or operator  (72ms)
      ✓ approving allows transfer from the approved party (94ms)
      ✓ transfering the approved NFT results in aproval reset for it (94ms)
      ✓ transfering the approved NFT results in aproval reset for it but no approval event (93ms)
      ✓ transfering the approved NFT again will fail (95ms)
      ✓ approval by operator works (122ms)
    ApprovalForAll
      ✓ approving all emit ApprovalForAll event (58ms)
      ✓ approving all update the approval status (60ms)
      ✓ unsetting approval for all should update the approval status (80ms)
      ✓ unsetting approval for all should emit ApprovalForAll event (78ms)
      ✓ approving for all allows transfer from the approved party (99ms)
      ✓ transfering one NFT do not results in aprovalForAll reset (92ms)
      ✓ approval for all does not grant approval on a transfered NFT (91ms)
      ✓ approval for all set before will work on a transfered NFT (121ms)
      ✓ approval for all allow to set individual nft approve (153ms)

  GameToken:ERC721
    non existing NFT
      ✓ transfering a non existing NFT fails (1335ms)
      ✓ tx balanceOf a zero owner fails (98ms)
      ✓ call balanceOf a zero owner fails (95ms)
      ✓ tx ownerOf a non existing NFT fails (94ms)
      ✓ call ownerOf a non existing NFT fails (93ms)
      ✓ tx getApproved a non existing NFT fails (90ms)
      ✓ call getApproved a non existing NFT fails (97ms)
    balance
      ✓ balance is zero for new user (95ms)
      ✓ balance return correct value (188ms)
    mint
      ✓ mint result in a transfer from 0 event (122ms)
      ✓ mint for gives correct owner (124ms)
    burn
      ✓ burn result in a transfer to 0 event (155ms)
      ✓ burn result in ownerOf throwing (147ms)
    batchTransfer
      ✓ batch transfer of same NFT ids should fails (96ms)
      ✓ batch transfer works (132ms)
    mandatory batchTransfer
      ✓ batch transfering to a contract that do not implements mandatory erc721 receiver but implement classic ERC721 receiver and reject should not fails (139ms)
      ✓ batch transfering to a contract that implements mandatory erc721 receiver (and signal it properly via 165) should fails if it reject it (105ms)
      ✓ batch transfering to a contract that do not accept erc721 token should fail (102ms)
      ✓ batch transfering to a contract that do not return the correct onERC721Received bytes shoudl fail (101ms)
      ✓ batch transfering to a contract that do not implemented mandatory receiver should not fail (137ms)
      ✓ batch transfering to a contract that return the correct onERC721Received bytes shoudl succeed (154ms)
    mandatory transfer
      ✓ transfering to a contract that do not implements mandatory erc721 receiver but implement classic ERC721 receiver and reject should not fails (134ms)
      ✓ transfering to a contract that implements mandatory erc721 receiver (and signal it properly via 165) should fails if it reject it (103ms)
      ✓ transfering to a contract that do not accept erc721 token should fail (101ms)
      ✓ transfering to a contract that do not return the correct onERC721Received bytes shoudl fail (103ms)
      ✓ transfering to a contract that do not implemented mandatory receiver should not fail (130ms)
      ✓ transfering to a contract that return the correct onERC721Received bytes shoudl succeed (323ms)
    safe batch transfer
      ✓ safe batch transfer of same NFT ids should fails (183ms)
      ✓ safe batch transfer works (132ms)
    transfer
      ✓ transfering one NFT results in one erc721 transfer event (116ms)
      ✓ transfering one NFT change to correct owner (119ms)
      ✓ transfering one NFT increase new owner balance (122ms)
      ✓ transfering one NFT decrease past owner balance (124ms)
      ✓ transfering from without approval should fails (97ms)
      ✓ transfering to zero address should fails (90ms)
      ✓ transfering to a contract that do not accept erc721 token should not fail (132ms)
    safeTransfer
      ✓ safe transfering one NFT results in one erc721 transfer event (115ms)
      ✓ safe transfering to zero address should fails (92ms)
      ✓ safe transfering one NFT change to correct owner (118ms)
      ✓ safe transfering from without approval should fails (90ms)
      ✓ safe transfering to a contract that do not accept erc721 token should fail (100ms)
      ✓ safe transfering to a contract that do not return the correct onERC721Received bytes shoudl fail (98ms)
      ✓ safe transfering to a contract that do not implemented onERC721Received should fail (100ms)
      ✓ safe transfering to a contract that return the correct onERC721Received bytes shoudl succeed (136ms)
    safeTransfer with empty bytes
      ✓ data:0x : safe transfering one NFT results in one erc721 transfer event (117ms)
      ✓ data:0x : safe transfering to zero address should fails (90ms)
      ✓ data:0x : safe transfering one NFT change to correct owner (118ms)
      ✓ data:0x : safe transfering from without approval should fails (89ms)
      ✓ data:0x : safe transfering to a contract that do not accept erc721 token should fail (99ms)
      ✓ data:0x : safe transfering to a contract that do not return the correct onERC721Received bytes shoudl fail (103ms)
      ✓ data:0x : safe transfering to a contract that do not implemented onERC721Received should fail (103ms)
      ✓ data:0x : safe transfering to a contract that return the correct onERC721Received bytes shoudl succeed (142ms)
    safeTransfer with data
      ✓ data:0xff56fe3422 : safe transfering one NFT results in one erc721 transfer event (117ms)
      ✓ data:0xff56fe3422 : safe transfering to zero address should fails (94ms)
      ✓ data:0xff56fe3422 : safe transfering one NFT change to correct owner (116ms)
      ✓ data:0xff56fe3422 : safe transfering from without approval should fails (91ms)
      ✓ data:0xff56fe3422 : safe transfering to a contract that do not accept erc721 token should fail (97ms)
      ✓ data:0xff56fe3422 : safe transfering to a contract that do not return the correct onERC721Received bytes shoudl fail (100ms)
      ✓ data:0xff56fe3422 : safe transfering to a contract that do not implemented onERC721Received should fail (100ms)
      ✓ data:0xff56fe3422 : safe transfering to a contract that return the correct onERC721Received bytes shoudl succeed (136ms)
    ERC165
      ✓ claim to support erc165 (95ms)
      ✓ claim to support base erc721 interface (88ms)
      ✓ claim to support erc721 metadata interface (91ms)
      ✓ does not claim to support random interface (92ms)
      ✓ does not claim to support the invalid interface (89ms)
    Approval
      ✓ approving emit Approval event (111ms)
      ✓ removing approval emit Approval event (137ms)
      ✓ approving update the approval status (111ms)
      ✓ cant approve if not owner or operator  (121ms)
      ✓ approving allows transfer from the approved party (146ms)
      ✓ transfering the approved NFT results in aproval reset for it (145ms)
      ✓ transfering the approved NFT results in aproval reset for it but no approval event (141ms)
      ✓ transfering the approved NFT again will fail (143ms)
      ✓ approval by operator works (170ms)
    ApprovalForAll
      ✓ approving all emit ApprovalForAll event (109ms)
      ✓ approving all update the approval status (114ms)
      ✓ unsetting approval for all should update the approval status (132ms)
      ✓ unsetting approval for all should emit ApprovalForAll event (130ms)
      ✓ approving for all allows transfer from the approved party (142ms)
      ✓ transfering one NFT do not results in aprovalForAll reset (133ms)
      ✓ approval for all does not grant approval on a transfered NFT (140ms)
      ✓ approval for all set before will work on a transfered NFT (168ms)
      ✓ approval for all allow to set individual nft approve (192ms)

  SafeMathWithRequire
{ loopCounter: 81 }
{ loopCounter: 173 }
    ✓ cbrt6
{ loopCounter: 47 }
{ loopCounter: 139 }
    ✓ cbrt3
```

```
{ loopCounter: 215 }
{ loopCounter: 469 }
    ✓ rt6_3

  LandWeightedSANDRewardPool computation
    ✓ computing contributions (486ms)

  MockSANDRewardPool
    ✓ Pool contains reward tokens (132ms)
    ✓ User with stakeTokens can stake
    ✓ User earnings for 0 NFTs match expected reward
    ✓ User earnings for 0 NFTs match expected reward with 1 stake
    ✓ User earnings for 0 NFTs match expected reward with 2 stakes
    ✓ User earnings for 0 NFTs match expected reward with 3 stakes
    ✓ User earnings for 0 NFTs match expected reward with 4 stakes
    ✓ User earnings for 0 NFTs match expected reward with 10 stakes (83ms)
    ✓ User earnings for 1 NFTs match expected reward
    ✓ User earnings for 1 NFTs match expected reward with 10 stakes (108ms)
    ✓ User earnings for 2 NFTs match expected reward
    ✓ User earnings for 3 NFTs match expected reward
    ✓ User earnings for 3 NFTs match expected reward with 10 stakes (124ms)
    ✓ User earnings for 89 NFTs match expected reward (397ms)
    ✓ User earnings for 89 NFTs match expected reward with 10 stakes (489ms)
    ✓ Multiple Users' earnings for 0 NFTs match expected reward: 2 users
    ✓ Multiple Users' earnings for 0 NFTs match expected reward: 2 users, 10 stakes each (157ms)
    ✓ Multiple Users' earnings for 0 NFTs match expected reward: 3 users, 1 stake each
    ✓ Multiple Users' earnings for 1 NFTs match expected reward: 2 users, 1 stake each
    ✓ Multiple Users' earnings for 1 NFTs match expected reward: 2 users, 10 stakes each (208ms)
    ✓ Multiple Users' earnings for 3 NFTs match expected reward: 2 users, 1 stake each (57ms)
    ✓ Multiple Users' earnings for 100 NFTs match expected reward: 2 users, 1 stake each (891ms)
    ✓ Staking with STAKE_AMOUNT plus an extra amount equivalent to 2 NFTs
    ✓ Earlier staker gets more rewards with same NFT amount - small NFT number (40ms)
    ✓ Earlier staker gets more rewards with same NFT amount - large NFT number (996ms)
    ✓ More lands give more rewards than earlier staker when NFT amounts are smaller (54ms)
    ✓ More lands do not give more rewards than earlier staker with large NFT amounts (976ms)
    ✓ rewardToken in pool is more than amount notified (155ms)
    ✓ rewardToken in pool is zero (153ms)
    ✓ rewardToken in pool is less than amount notified
    ✓ the call to notifyRewardAmount is made after users first call stake (165ms)
    ✓ user is earning rewards and pool is notified for a second time before end of current reward period (148ms)

  ActualSANDRewardPool
    ✓ Contract should exist (583ms)
    ✓ Pool contains reward tokens
    ✓ User with stakeTokens can stake
    ✓ User can earn rewardTokens if pool has been notified of reward
    ✓ admin can notifyRewardAmount and start a new reward process (without sending more reward tokens)
    ✓ User cannot earn rewardTokens if they stake after the end time
    ✓ User earns full reward amount if they are the only staker after 1 day
    ✓ User earns full reward amount if they are the only staker after 29 days
    ✓ User with 0 LAND earns correct reward amount
    ✓ User with 0 LAND earns correct reward amount - smaller stake
    ✓ User with 1 LAND earns correct reward amount
    ✓ User with 3 LANDs earns correct reward amount (43ms)
    ✓ User with 10 LANDs earns correct reward amount (77ms)
    ✓ User can withdraw some stakeTokens after several amounts have been staked
    ✓ First user can withdraw their stakeTokens
    ✓ User can withdraw all stakeTokens after several amounts have been staked (40ms)
    ✓ First user can claim their reward - no NFTs
    ✓ First user can claim their reward - has NFTs (83ms)
    ✓ A user can claim their reward after multiple stakes (124ms)
    ✓ First user can exit the pool
    ✓ A user can exit the pool after multiple stakes (43ms)
    ✓ A user with NFTs can exit the pool after multiple stakes (111ms)

  Catalyst_EPIC
    ✓ transfering from users[0] to users[1] should adjust their balance accordingly (517ms)
    ✓ transfering from users[0] more token that it owns should fails
    ✓ transfering to address zero should fails
    ✓ transfering to address(this) should fail
    ✓ transfering from users[0] to users[1] by users[0] should adjust their balance accordingly
    ✓ transfering from users[0] by users[1] should fails
    ✓ transfering from users[0] to users[1] should trigger a transfer event
    ✓ transfering from users[0] to users[1] by operator after approval, should adjust their balance accordingly (62ms)
    ✓ transfering from users[0] to users[1] by operator after approval and approval reset, should fail (59ms)
    ✓ transfering from users[0] to users[1] by operator after approval, should adjust the operator alowance accordingly (39ms)
    ✓ transfering from users[0] to users[1] by operator after max approval (2**256-1), should NOT adjust the operator allowance
    ✓ transfering from users[0] to users[1] by operator after approval, but without enough allowance, should fails
    ✓ transfering from users[0] by operators without pre-approval should fails
    ✓ approving operator should trigger a Approval event
    ✓ disapproving operator (allowance to zero) should trigger a Approval event (46ms)
    ✓ approve to address zero should fails

  Gem_POWER
    ✓ transfering from users[0] to users[1] should adjust their balance accordingly (556ms)
    ✓ transfering from users[0] more token that it owns should fails
    ✓ transfering to address zero should fails
    ✓ transfering to address(this) should fail
    ✓ transfering from users[0] to users[1] by users[0] should adjust their balance accordingly
    ✓ transfering from users[0] by users[1] should fails
    ✓ transfering from users[0] to users[1] should trigger a transfer event
    ✓ transfering from users[0] to users[1] by operator after approval, should adjust their balance accordingly (59ms)
    ✓ transfering from users[0] to users[1] by operator after approval and approval reset, should fail (40ms)
    ✓ transfering from users[0] to users[1] by operator after approval, should adjust the operator alowance accordingly
    ✓ transfering from users[0] to users[1] by operator after max approval (2**256-1), should NOT adjust the operator allowance
    ✓ transfering from users[0] to users[1] by operator after approval, but without enough allowance, should fails
    ✓ transfering from users[0] by operators without pre-approval should fails
    ✓ approving operator should trigger a Approval event
    ✓ disapproving operator (allowance to zero) should trigger a Approval event (39ms)
    ✓ approve to address zero should fails

  LandBaseToken:ERC721
    non existing NFT
      ✓ transfering a non existing NFT fails (222ms)
      ✓ tx balanceOf a zero owner fails
      ✓ call balanceOf a zero owner fails
      ✓ tx ownerOf a non existing NFT fails (40ms)
      ✓ call ownerOf a non existing NFT fails
      ✓ tx getApproved a non existing NFT fails
      ✓ call getApproved a non existing NFT fails
    balance
      ✓ balance is zero for new user
      ✓ balance return correct value (85ms)
    mint
      ✓ mint result in a transfer from 0 event
      ✓ mint for gives correct owner (41ms)
    burn
      ✓ burn result in a transfer to 0 event (67ms)
      ✓ burn result in ownerOf throwing (69ms)
    batchTransfer
      ✓ batch transfer of same NFT ids should fails
      ✓ batch transfer works (81ms)
    mandatory batchTransfer
      ✓ batch transfering to a contract that do not implements mandatory erc721 receiver but implement classic ERC721 receiver and reject should not fails (81ms)
      ✓ batch transfering to a contract that implements mandatory erc721 receiver (and signal it properly via 165) should fails if it reject it (39ms)
      ✓ batch transfering to a contract that do not accept erc721 token should fail (39ms)
      ✓ batch transfering to a contract that do not return the correct onERC721Received bytes shoudl fail (43ms)
      ✓ batch transfering to a contract that do not implemented mandatory receiver should not fail (81ms)
      ✓ batch transfering to a contract that return the correct onERC721Received bytes shoudl succeed (54ms)
    mandatory transfer
      ✓ transfering to a contract that do not implements mandatory erc721 receiver but implement classic ERC721 receiver and reject should not fails (74ms)
      ✓ transfering to a contract that implements mandatory erc721 receiver (and signal it properly via 165) should fails if it reject it
      ✓ transfering to a contract that do not accept erc721 token should fail (39ms)
      ✓ transfering to a contract that do not return the correct onERC721Received bytes shoudl fail (41ms)
      ✓ transfering to a contract that do not implemented mandatory receiver should not fail (76ms)
      ✓ transfering to a contract that return the correct onERC721Received bytes shoudl succeed (53ms)
    safe batch transfer
      ✓ safe batch transfer of same NFT ids should fails
      ✓ safe batch transfer works (79ms)
    transfer
      ✓ transfering one NFT results in one erc721 transfer event
      ✓ transfering one NFT change to correct owner (41ms)
      ✓ transfering one NFT increase new owner balance (43ms)
      ✓ transfering one NFT decrease past owner balance (48ms)
      ✓ transfering from without approval should fails
      ✓ transfering to zero address should fails
      ✓ transfering to a contract that do not accept erc721 token should not fail (71ms)
    safeTransfer
      ✓ safe transfering one NFT results in one erc721 transfer event (38ms)
      ✓ safe transfering to zero address should fails
      ✓ safe transfering one NFT change to correct owner (39ms)
      ✓ safe transfering from without approval should fails
      ✓ safe transfering to a contract that do not accept erc721 token should fail (39ms)
      ✓ safe transfering to a contract that do not return the correct onERC721Received bytes shoudl fail (38ms)
      ✓ safe transfering to a contract that do not implemented onERC721Received should fail (38ms)
      ✓ safe transfering to a contract that return the correct onERC721Received bytes shoudl succeed (53ms)
    safeTransfer with empty bytes
      ✓ data:0x : safe transfering one NFT results in one erc721 transfer event (41ms)
      ✓ data:0x : safe transfering to zero address should fails
      ✓ data:0x : safe transfering one NFT change to correct owner (41ms)
      ✓ data:0x : safe transfering from without approval should fails
      ✓ data:0x : safe transfering to a contract that do not accept erc721 token should fail (43ms)
      ✓ data:0x : safe transfering to a contract that do not return the correct onERC721Received bytes shoudl fail (52ms)
      ✓ data:0x : safe transfering to a contract that do not implemented onERC721Received should fail (40ms)
      ✓ data:0x : safe transfering to a contract that return the correct onERC721Received bytes shoudl succeed (58ms)
    safeTransfer with data
```

```
      ✓ data:0xff56fe3422 : safe transfering one NFT results in one erc721 transfer event (49ms)
      ✓ data:0xff56fe3422 : safe transfering to zero address should fails
      ✓ data:0xff56fe3422 : safe transfering one NFT change to correct owner (46ms)
      ✓ data:0xff56fe3422 : safe transfering from without approval should fails
      ✓ data:0xff56fe3422 : safe transfering to a contract that do not accept erc721 token should fail (40ms)
      ✓ data:0xff56fe3422 : safe transfering to a contract that do not return the correct onERC721Received bytes shoudl fail (38ms)
      ✓ data:0xff56fe3422 : safe transfering to a contract that do not implemented onERC721Received should fail (39ms)
      ✓ data:0xff56fe3422 : safe transfering to a contract that return the correct onERC721Received bytes shoudl succeed (55ms)
    ERC165
      ✓ claim to support erc165
      ✓ claim to support base erc721 interface
      ✓ claim to support erc721 metadata interface
      ✓ does not claim to support random interface
      ✓ does not claim to support the invalid interface
    Approval
      ✓ approving emit Approval event (39ms)
      ✓ removing approval emit Approval event (55ms)
      ✓ approving update the approval status (45ms)
      ✓ cant approve if not owner or operator  (44ms)
      ✓ approving allows transfer from the approved party (54ms)
      ✓ transfering the approved NFT results in aproval reset for it (56ms)
      ✓ transfering the approved NFT results in aproval reset for it but no approval event (50ms)
      ✓ transfering the approved NFT again will fail (57ms)
      ✓ approval by operator works (85ms)
    ApprovalForAll
      ✓ approving all emit ApprovalForAll event
      ✓ approving all update the approval status
      ✓ unsetting approval for all should update the approval status (61ms)
      ✓ unsetting approval for all should emit ApprovalForAll event (60ms)
      ✓ approving for all allows transfer from the approved party (53ms)
      ✓ transfering one NFT do not results in aprovalForAll reset (791ms)
      ✓ approval for all does not grant approval on a transfered NFT (58ms)
      ✓ approval for all set before will work on a transfered NFT (80ms)
      ✓ approval for all allow to set individual nft approve (97ms)

  Asset.sol
    ✓ user sending asset to itself keep the same balance (87ms)
    ✓ can transfer assets
    ✓ user batch sending asset to itself keep the same balance
    ✓ user batch sending in series whose total is more than its balance
    ✓ user batch sending more asset that it owns should fails
    ✓ can get the chainIndex from the tokenId
    ✓ can get the URI for an NFT
    ✓ can get the URI for a FT
    ✓ fails get the URI for an invalid tokeId
    ✓ can burn ERC1155 asset
    ✓ can burn ERC721 asset
    Asset: MetaTransactions
      ✓ can transfer by metaTx
      ✓ fails to transfer someone else token by metaTx
      ✓ can batch-transfer by metaTx (39ms)

  assetSignedAuctionWithAuth
    ✓ should be able to set fee (1110ms)
    ✓ should fail setting fee - no admin
    ✓ should fail is buyer == seller (41ms)
    ✓ should fail is ids.length != amounts.length
    ✓ should fail - insuficient amount
    ✓ should be able to claim seller offer in ETH (47ms)
    ✓ should NOT be able to claim offer if signature mismatches
    ✓ should NOT be able to claim offer with invalid backend signature
    ✓ should be able to claim seller offer in SAND (56ms)
    ✓ should be able to claim seller offer with basic signature (51ms)
    ✓ should be able to cancel offer
    ✓ should NOT be able to claim offer without sending ETH
    ✓ should NOT be able to claim offer without enough SAND (46ms)
    ✓ should NOT be able to claim offer if it did not start yet
    ✓ should NOT be able to claim offer if it already ended

  Asset_Giveaway
    ✓ User cannot claim when test contract holds zero assets (73ms)
    ✓ User can claim allocated multiple assets for multiple assetIds from Giveaway contract (170ms)
    ✓ Claimed Event is emitted for successful claim
    ✓ User can claim allocated single asset for single assetId from Giveaway contract
    ✓ User tries to claim the wrong amount of an assetID
    ✓ User cannot claim their assets more than once
    ✓ User cannot claim assets from Giveaway contract if destination is not the reserved address
    ✓ User cannot claim assets from Giveaway contract to destination zeroAddress
    ✓ User cannot claim assets from Giveaway contract with incorrect asset param
    ✓ User can claim allocated multiple assets for multiple assetIds from alternate address (155ms)
    ✓ merkleRoot cannot be set twice (52ms)
    ✓ merkleRoot can only be set by admin

  GAS:Asset_Giveaway_1:Claiming
    ✓ 1 claim (86ms)
    ✓ 10 claims (88ms)
    ✓ 4000 claims (773ms)
    ✓ 10000 claims (1782ms)
{
  "Gas per claim - 1 claim total": 112331,
  "Gas per claim - 10 claims total": 115008,
  "Gas per claim - 4000 claims total": 122100,
  "Gas per claim - 10000 claims total": 123886
}

  MerkleTree_assets
    ✓ should validate the data

  SignedGiveaway.sol
    initialization
      ✓ interfaces (307ms)
    roles
      ✓ admin
      ✓ signer
    claim
      ✓ should be able to claim sand
      ✓ should fail to claim the same id twice
      ✓ should fail to claim if the signature is wrong
      ✓ should fail to mint if the signer is invalid
      ✓ claim with metaTX trusted forwarder
    revoke
      ✓ should fail to revoke if not admin
      ✓ should fail to claim if the id was revoked
    pause
      ✓ should fail to pause if not admin
      ✓ should fail to unpause if not admin
      ✓ should fail to claim if paused
      ✓ should be able to claim sand after pause/unpause (38ms)
    coverage
      ✓ a valid signature must verify correctly
      ✓ check the domain separator

  SafeMathWithRequire.sol library via MockSafeMathWithRequire.sol
    ✓ sqrt6, sqrt multiplied by 1e6
    ✓ sqrt3, sqrt multiplied by 1e3
    ✓ cbrt6, cube root multiplied by 1e6
    ✓ cbrt3, cube root multiplied by 1e3

  ERC20RewardPool main contract tests
    roles
      ✓ admin should be able to call setContributionRules (624ms)
      ✓ other should fail to call setContributionRules
      ✓ admin should be able to call setRewardToken
      ✓ other should fail to call setRewardToken
      ✓ admin should be able to call setStakeToken
      ✓ other should fail to call setStakeToken
      ✓ admin should be able to call setRewardCalculator
      ✓ other should fail to call setRewardCalculator
      ✓ admin should be able to call pause & unpause
      ✓ other should fail to call pause & unpause
      ✓ admin cant renounce ownership
      ✓ recoverFunds should fail if contract is not paused
      ✓ admin should be able to call recoverFunds if contract is paused (39ms)
      ✓ other should fail to call recoverFunds
      ✓ recoverFunds must fail with address zero
      ✓ contract should have enough funds to replace the stakeToken (47ms)
      ✓ contract should have enough funds to replace the rewardToken
    reward distribution
      only one user
        ✓ reward before stake (135ms)
        ✓ stake before rewards (141ms)
        ✓ stake->withdraw so total contribution == 0, stake again (118ms)
        ✓ stake->withdraw so total contribution == 0, stake again with some rewards (180ms)
      two users
        ✓ reward before stake (133ms)
        ✓ user1 stake before rewards (149ms)
        ✓ user2 stake before rewards (152ms)
      10 users
        ✓ reward before stake (1155ms)
    contribution calculation
      ✓ initial (548ms)
      ✓ computeContribution after the user change his contribution (314ms)
      ✓ computeContributionInBatch after the user change his contribution (424ms)
    contribution calculation with rewards
      ✓ initial (614ms)
      ✓ computeContribution after users change his contribution (508ms)
```

```
     ✓ computeContributionInBatch after the user change his contribution (348ms)
  trusted forwarder and meta-tx
     ✓ should fail to set the trusted forwarder if not admin
     ✓ should success to set the trusted forwarder if a valid contract and admin
     ✓ setReward with meta-tx
     ✓ stake with meta-tx (47ms)
     ✓ withdraw with meta-tx (57ms)

ContributionRules
  roles
     ✓ admin should be able to call setERC721MultiplierList (469ms)
     ✓ setERC721MultiplierList above the limits should fail (104ms)
     ✓ admin should be able to call setERC1155MultiplierList
     ✓ setERC1155MultiplierList above the limits should fail (135ms)
     ✓ admin should be able to call deleteERC721MultiplierList
     ✓ admin should be able to call deleteERC1155MultiplierList
     ✓ admin should be able to call setERC721MultiplierLimit
     ✓ admin should be able to call setERC1155MultiplierLimit
  compute contribution
     ✓ 0 ERC721 - 0 ERC1155
     ✓ 1 ERC721 balanceOf - 0 ERC1155 (41ms)
     ✓ 4 ERC721 balanceOf - 0 ERC1155 (50ms)
     ✓ 1 ERC721 balanceOf - 1 ERC1155 (55ms)
     ✓ 1 ERC721 balanceOf - 2 ERC1155 ids (63ms)
     ✓ 1 ERC721 id - 2 ERC1155 ids (43ms)
     ✓ 2 ERC721 ids - 2 ERC1155 ids (53ms)
  multiplier limit
     ✓ should limit ERC721 multiplier at 15% (77ms)
     ✓ should limit ERC1155 multiplier at 15%
     ✓ should return MaxGlobalMultiplier
     ✓ should not be able to set setERC721MultiplierLimit > 1000
     ✓ should not be able to set setERC1155MultiplierLimit > 1000

ERC20RewardPool Lock Rules
   ✓ admin should be able to call setTimelockClaim (585ms)
   ✓ other should fail to call setTimelockClaim
   ✓ should fail to setTimelockClaim above the limit
   ✓ user can only get his rewards after lockTimeMS (109ms)
   ✓ we can disable lockTimeMS check by setting it to zero (64ms)
   ✓ admin should be able to call setTimelockDeposit
   ✓ other should fail to call setTimelockDeposit
   ✓ should fail to setTimelockDeposit above the limit
   ✓ user should wait to deposit(stake) again
   ✓ admin should be able to call setTimeLockWithdraw
   ✓ other should fail to call setTimeLockWithdraw
   ✓ should fail to setTimeLockWithdraw above the limit
   ✓ user should wait to withdraw again and exit (75ms)
   ✓ should fail to setAmountLockClaim above the limit
   ✓ should be able to claim only amount allowed or if check is disabled (68ms)

Requirementsules
  roles
     ✓ admin should be able to call setMaxStakeOverall
     ✓ admin should be able to call setERC721RequirementList
     ✓ admin should be able to call setERC1155RequirementList
     ✓ admin should be able to call deleteERC721RequirementList
     ✓ admin should be able to call deleteERC1155RequirementList
  Max/Min Stake
     ✓ getERC721MaxStake should return correct values - balanceOf (38ms)
     ✓ getERC721MaxStake should return correct values - id (38ms)
     ✓ getERC721MaxStake should return correct values - id & balanceOf (44ms)
     ✓ checkERC721MinStake should fail - balanceOf (54ms)
     ✓ checkERC721MaxStake should return correct value (53ms)
     ✓ checkERC1155MaxStake should return correct value (61ms)
  Max Stake Calculator
     ✓ ERC721 balanceOf and ERC1155 (58ms)
     ✓ ERC721 balanceOf and No ERC1155 (53ms)
     ✓ ERC721 balanceId and ERC1155 (50ms)
     ✓ No ERC721 and ERC1155 (49ms)
     ✓ maxStakeOverall should cap maxStake (48ms)
  Stake
     ✓ user should be able to stake (65ms)
     ✓ stake should fail - ERC721 balanceId  (43ms)
     ✓ stake should fail - ERC721 balanceOf  (48ms)
     ✓ stake should fail - ERC1155 balanceId  (47ms)
     ✓ stake should fail - maxAllowed  (51ms)

LandContributionCalculator
  roles
     ✓ admin should be able to call setNFTMultiplierToken (144ms)
     ✓ others should fail to call setNFTMultiplierToken (49ms)
  calculation
     ✓ zero lands
     ✓ 1 lands
     ✓ 2 lands
     ✓ 3 lands
     ✓ 4 lands (39ms)
     ✓ 5 lands, to high to be minted
     ✓ 10 lands, to high to be minted
     ✓ 20 lands, to high to be minted (41ms)
     ✓ 50 lands, to high to be minted
     ✓ 100 lands, to high to be minted
     ✓ 408 lands, to high to be minted
     ✓ 166464 lands, to high to be minted
     ✓ 67917312 lands, to high to be minted (41ms)

LandOwnerContributionCalculator
  roles
     ✓ admin should be able to call setNFTMultiplierToken (142ms)
     ✓ others should fail to call setNFTMultiplierToken (52ms)
  calculation
     ✓ users without lands get zero contributions
     ✓ 1 lands
     ✓ 2 lands
     ✓ 3 lands
     ✓ 4 lands

PeriodicRewardCalculator
   ✓ only Admin can call setDuration (72ms)
   ✓ calling setDuration should update campaign duration
   ✓ calling setDuration during the campaing should fail
  roles
     ✓ reward pool should be able to call restartRewards
     ✓ others should fail to call restartRewards
     ✓ reward distribution should be able to call notifyRewardAmount
     ✓ other should fail to call notifyRewardAmount
     ✓ reward distribution should be able to call setSavedRewards
     ✓ other should fail to call setSavedRewards
  should be no rewards on initialization
     ✓ startup
     ✓ restart call
  reward distribution
     ✓ setup: we use the rate, so REWARDS must be multiple of duration (or leftover + reward if we add in the middle)
     ✓ we distribute rewards linearly (283ms)
     ✓ if restart is called (with contribution!=0) then rewards starts from zero again (218ms)
     ✓ calling notifyRewardAmount in the middle of the distribution will distribute the remaining + what was added (290ms)
     ✓ calling notifyRewardAmount after the distribution will distribute both amounts (292ms)

TwoPeriodsRewardCalculator
   ✓ startup (71ms)
  roles
     ✓ reward pool should be able to call restartRewards
     ✓ others should fail to call restartRewards
     ✓ reward distribution should be able to call (c) => c.runCampaign(12345678, 9876)
     ✓ other should fail to call (c) => c.runCampaign(12345678, 9876)
     ✓ reward distribution should be able to call (c) => c.setInitialCampaign(12345678, 9876)
     ✓ other should fail to call (c) => c.setInitialCampaign(12345678, 9876)
     ✓ reward distribution should be able to call (c) => __awaiter(this, void 0, void 0, function* () {
        yield c.setInitialCampaign(123, 1234);
        return c.updateNextCampaign(12345678, 9876);
    })
     ✓ other should fail to call (c) => __awaiter(this, void 0, void 0, function* () {
        yield c.setInitialCampaign(123, 1234);
        return c.updateNextCampaign(12345678, 9876);
    })
  setup restrictions
     ✓ should fail to set initial campaign if campaign is running
     ✓ should fail to set next campaign if no campaign is running
     ✓ should fail to update current campaign if no campaign is running
     ✓ run campaign always works (45ms)
  reward distribution
     ✓ run an initial campaign alone
     ✓ run initial and next campaign (75ms)
     ✓ run next campaign after the initial one finished (117ms)
  restart reward
     ✓ before everything
     ✓ in middle of the first campaign
     ✓ in middle of the second campaign
     ✓ after everything
     ✓ intermixed (50ms)

SandRewardPool
   ✓ last time reward application should match the duration (692ms)
   ✓ total supply is at first empty
   ✓ staking should update the reward balance, supply and staking token balance
   ✓ withdraw should update the reward balance, supply and staking token (41ms)
```

```
     ✓ reward per token should be 0 if total supply is 0
     ✓ reward per token calculation
     ✓ earned calculation
     ✓ get reward should transfer the reward and emit an event (56ms)
     ✓ exiting should withdraw and transfer the reward (45ms)
     ✓ pool contains reward tokens
     ✓ user can earn reward tokens if pool has been notified of reward
     ✓ admin can notify to start a new reward process (without sending more reward tokens)
     ✓ user cannot earn rewardTokens if they stake after the end time
     ✓ user earns full reward amount if there is only one staker after 1 day(s)
     ✓ user earns full reward amount if there is only one staker after 27 day(s)
     ✓ User with 0 LAND earns correct reward amount
     ✓ User with 0 LAND earns correct reward amount - smaller stake
     ✓ User with 1 LAND(s) earns correct reward amount (64ms)
     ✓ User with 3 LAND(s) earns correct reward amount (85ms)
     ✓ User with 10 LAND(s) earns correct reward amount (138ms)
     ✓ User can withdraw some stakeTokens after several amounts have been staked (42ms)
     ✓ First user can claim their reward - no NFTs (47ms)
     ✓ First user can claim their reward - has NFTs (154ms)
     ✓ A user can claim their reward after multiple stakes (203ms)
     ✓ First user can exit the pool (45ms)
     ✓ A user can exit the pool after multiple stakes (66ms)
     ✓ A user with NFTs can exit the pool after multiple stakes (219ms)
     ✓ Change externals contracts
     ✓ user earnings for 0 NFT(s) match expected reward with 1 stake(s)
     ✓ user earnings for 0 NFT(s) match expected reward with 2 stake(s) (41ms)
     ✓ user earnings for 0 NFT(s) match expected reward with 4 stake(s) (66ms)
     ✓ user earnings for 0 NFT(s) match expected reward with 10 stake(s) (138ms)
     ✓ user earnings for 1 NFT(s) match expected reward with 1 stake(s) (48ms)
     ✓ user earnings for 1 NFT(s) match expected reward with 10 stake(s) (259ms)
     ✓ user earnings for 2 NFT(s) match expected reward with 1 stake(s) (62ms)
     ✓ user earnings for 3 NFT(s) match expected reward with 1 stake(s) (67ms)
     ✓ user earnings for 3 NFT(s) match expected reward with 10 stake(s) (953ms)
     ✓ user earnings for 89 NFT(s) match expected reward with 1 stake(s) (749ms)
     ✓ user earnings for 89 NFT(s) match expected reward with 10 stake(s) (969ms)
     ✓ Multiple Users' earnings for 0 NFTs match expected reward: 2 users, 10 stake each (253ms)
     ✓ Multiple Users' earnings for 0 NFTs match expected reward: 3 users, 1 stake each (59ms)
     ✓ Multiple Users' earnings for 1 NFTs match expected reward: 2 users, 1 stake each (79ms)
     ✓ Multiple Users' earnings for 1 NFTs match expected reward: 2 users, 10 stake each (83ms)
     ✓ Multiple Users' earnings for 3 NFTs match expected reward: 2 users, 1 stake each (116ms)
     ✓ Multiple Users' earnings for 100 NFTs match expected reward: 2 users, 1 stake each (1541ms)
     ✓ Staking with STAKE_AMOUNT plus an extra amount equivalent to 2 NFTs (41ms)
     ✓ Earlier staker gets more rewards with same NFT amount - small NFT number (82ms)
     ✓ Earlier staker gets more rewards with same NFT amount - large NFT number (1514ms)
     ✓ More lands give more rewards than earlier staker when NFT amounts are smaller (93ms)
     ✓ More lands do not give more rewards than earlier staker with large NFT amounts (1545ms)
     ✓ rewardToken in pool is more than amount notified
     ✓ rewardToken in pool is zero
     ✓ rewardToken in pool is less than amount notified
     ✓ the call to notifyRewardAmount is made after users first call stake
     ✓ user is earning rewards and pool is notified for a second time before end of current reward period
     ✓ Multiplier & reward are correct (115ms)
    - THIS IS FALSE, EVERYBODY CAN DO IT: Only sender or reward distribution can compute sender's account

  LandOwnersSandRewardPool
     ✓ users with land should be able to stake (742ms)
     ✓ users without land should revert
     ✓ if a user sells his land we can recompute the contribution (139ms)

  new SandRewardPool main contract tests
    roles
       ✓ admin should be able to call setContributionCalculator (234ms)
       ✓ other should fail to call setContributionCalculator
       ✓ admin should be able to call setRewardToken
       ✓ other should fail to call setRewardToken
       ✓ admin should be able to call setStakeToken
       ✓ other should fail to call setStakeToken
       ✓ admin should be able to call setRewardCalculator
       ✓ other should fail to call setRewardCalculator
       ✓ admin should be able to call recoverFunds
       ✓ other should fail to call recoverFunds
       ✓ recoverFunds must fail with address zero
    reward distribution
      only one user
         ✓ reward before stake (103ms)
         ✓ stake before rewards (103ms)
         ✓ stake->withdraw so total contribution == 0, stake again (79ms)
         ✓ stake->withdraw so total contribution == 0, stake again with some rewards (111ms)
      two users
         ✓ reward before stake (104ms)
         ✓ user1 stake before rewards (115ms)
         ✓ user2 stake before rewards (115ms)
      10 users
         ✓ reward before stake (383ms)
    contribution calculation
       ✓ initial (116ms)
       ✓ computeContribution after the user change his contribution (233ms)
       ✓ computeContributionInBatch after the user change his contribution (215ms)
    contribution calculation with rewards
       ✓ initial (304ms)
       ✓ computeContribution after users change his contribution (471ms)
       ✓ computeContributionInBatch after the user change his contribution (298ms)
    trusted forwarder and meta-tx
       ✓ should fail to set the trusted forwarder if not admin
       ✓ should success to set the trusted forwarder if admin
       ✓ setReward with meta-tx
       ✓ stake with meta-tx
       ✓ withdraw with meta-tx (44ms)

  new SandRewardPool anti compound tests
     ✓ user can only get his rewards after lockTimeMS (91ms)
     ✓ we can disable lockTimeMS check by setting it to zero (51ms)
    roles
       ✓ admin should be able to call setAntiCompoundLockPeriod
       ✓ other should fail to call setAntiCompoundLockPeriod

  ERC677Token
     ✓ Transfering tokens to ERC677Receiver contract should emit an OnTokenTransferEvent event (617ms)
     ✓ Transfering tokens to EOA
     ✓ Transfering tokens to a non receiver contract should fail
     ✓ Transfering tokens to a contract with fallback function should succeed

  use withSnapshot to keep your testing environment clean
     ✓ withSnapshot doesn't care about what happen before (226ms)

  Faucet
     ✓ Send cannot exceed Faucet limit amout (229ms)
     ✓ Send cannot be executed twice without awaiting (43ms)
     ✓ Send succeded for correct asked amount
     ✓ Retrieve succeded for deployer
     ✓ Retrieve succeded for deployer with any address
     ✓ Retrieve fail for user that is not deployer
     ✓ setPeriod succeed for deployer
     ✓ setPeriod fail for user that is not deployer
     ✓ setLimit succeed for deployer
     ✓ Send with new limit succeed after limit update. (38ms)
     ✓ setLimit fail for user that is not deployer

  GameMinter
    GameMinter: Calling Directly
       ✓ should fail to create GAME if user has insufficient SAND
       ✓ should allow anyone to create a game
       ✓ should allow owner to add assets
       ✓ should charge a fee when owner adds assets
       ✓ should allow editor to add assets
       ✓ should allow owner to remove assets
       ✓ should allow editor to remove assets
       ✓ should fail if not authorized to add assets
       ✓ should fail to modify GAME if user has insufficient SAND
       ✓ should fail if not authorized to remove assets
       ✓ should fail if not authorized to set GAME URI
       ✓ allows GAME owner to set GAME URI
       ✓ allows GAME editor to set GAME URI
    GameMinter: Sandbox MetaTXs
       ✓ should allow anyone to create a game via MetaTx
       ✓ should allow GAME Owner to add assets via MetaTx (40ms)
       ✓ should allow GAME Owner to remove assets via MetaTx
       ✓ should allow GAME Owner to set URI via MetaTx (40ms)
       ✓ should allow GAME Editor to add assets via MetaTx
       ✓ should allow GAME Editor to remove assets via MetaTx
       ✓ should allow GAME Editor to set URI via MetaTx (42ms)

  GameToken
    GameToken: Minting GAMEs
       ✓ can update the GameMinter address
       ✓ Minter can create GAMEs when _Minter is set
       ✓ should revert if trying to reuse a baseId
       ✓ gameId contains creator, randomId, chainIndex & version data
       ✓ can get the storageId for a GAME
       ✓ can get the chainIndex for a GAME
       ✓ reverts if non-minter trys to mint Game when _Minter is set
      GameToken: Mint With Assets
         ✓ fails to create if "to" address is the gameToken contract
         ✓ fails to add ERC1155 tokens to the game if Operator != GAME contract
         ✓ fails to add ERC1155 token batch to the game if Operator != GAME contract (55ms)
         ✓ can mint Games with single Asset (50ms)
```

```
        ✓ can mint Games with many Assets (79ms)
        ✓ should fail if length of assetIds and values dont match
      GameToken: Modifying GAMEs
        ✓ should allow the owner to add game editors
        ✓ should allow the owner to remove game editors
        ✓ should revert if non-owner trys to set Game Editors
        ✓ Minter can add single Asset (72ms)
        ✓ should bump the version number in the gameId
        ✓ Minter can add multiple Assets (78ms)
        ✓ Minter can remove single Asset
        ✓ fails when removing more assets than the game contains
        ✓ Minter can remove multiple Assets (45ms)
        ✓ Game token should acurately track token balances for owners
    GameToken: Transferring GAMEs
      ✓ current owner can transfer ownership of a GAME
      ✓ can transfer creatorship of a GAME
      ✓ transfer creatorship should revert for a non existing game
      ✓ can transfer creatorship of a GAME back to original creator
      ✓ should fail if non-owner trys to transfer a GAME
      ✓ transfer creatorship should revert for a burned game
    GameToken: MetaData
      ✓ can get the ERC721 token contract name
      ✓ can get the ERC721 token contract symbol
      ✓ can get the tokenURI
      ✓ Minter can set the tokenURI
      ✓ should revert if ownerOf == address(0)
      ✓ should revert if not Minter
      ✓ should be able to retrieve the creator address from the gameId
    GameToken: Destroying Games
      ✓ fails if "from" != game owner
      ✓ fails if sender != game owner and not metatx
      GameToken: burnAndRecover
        ✓ fails if "to" == address(0)
        ✓ fails to destroy if "to" == Game Token contract
        ✓ fails if "from" != game owner
        ✓ fails if sender != game owner and not metatx
        ✓ can destroy GAME and recover assets in 1 tx if not too many assets (38ms)
        ✓ creatorOf() should should still return original creator
        ✓ game should no longer exist
      GameToken: Destroy... then Recover
        ✓ fails to recover if the GAME token has not been burnt
        ✓ can destroy without transfer of assets
        ✓ fails to recover if "to" address is the gameToken contract
        ✓ fails to recover assets if caller is not from or validMetaTx
        ✓ can recover remaining assets from burnt GAME in batches (48ms)
    GameToken: Token Immutability
      ✓ should store the creator address, subID & version in the gameId
      ✓ should consider future versions of gameIds as invalid
      ✓ should update version when changes are made
      ✓ should use baseId (creator address + subId) to map to game Assets
    GameToken: MetaTransactions
      ✓ can get isTrustedForwarder
      ✓ can call setGameEditor via metaTx
      ✓ can call burnFrom via metaTx (69ms)
      ✓ can call recoverAssets via metaTx (39ms)
      ✓ can call transferCreatorship via metaTx

  AuthValidator
    ✓ signature should be valid (168ms)
    ✓ signature should be invalid

  EstateSaleWithAuth
    ✓ should be able to purchase with valid signature (1587ms)
    ✓ should NOT be able to purchase with invalid signature
    ✓ should be able to purchase through sand contract

  GAS:Multi_Giveaway_1:Claiming
    ✓ 1 claim (1774ms)
    ✓ 10 claims (352ms)
    ✓ 4000 claims (3327ms)
    ✓ 10000 claims (7854ms)
{
  "Gas per claim - 1 claim total": 200591,
  "Gas per claim - 10 claims total": 203257,
  "Gas per claim - 4000 claims total": 210383,
  "Gas per claim - 10000 claims total": 212202
}

  MerkleTree_multi
    ✓ should validate the data

  Multi_Giveaway
    Multi_Giveaway_common_functionality
      ✓ Admin has the correct role (237ms)
      ✓ Admin can add a new giveaway
      ✓ Cannot add a new giveaway if not admin
      ✓ User can get their claimed status (229ms)
      ✓ Claimed status is correctly updated after allocated tokens are claimed - 2 claims of 2 claimed (632ms)
      ✓ Claimed status is correctly updated after allocated tokens are claimed - 1 claim of 2 claimed (38ms)
      ✓ MultiGiveaway contract returns ERC721 received (214ms)
      ✓ MultiGiveaway contract returns ERC721 Batch received
      ✓ MultiGiveaway contract returns ERC1155 received for supply 1
      ✓ MultiGiveaway contract returns ERC1155 received
      ✓ MultiGiveaway contract returns ERC1155 Batch received
    Multi_Giveaway_single_giveaway
      ✓ User cannot claim when test contract holds no tokens
      ✓ User cannot claim sand when contract does not hold any (439ms)
      ✓ User can claim allocated multiple tokens from Giveaway contract (498ms)
Number of assets: 64 ; Gas used: 2048530
      ✓ User can claim allocated 64 tokens from Giveaway contract (1716ms)
      ✓ Claimed Event is emitted for successful claim (457ms)
      ✓ User can claim allocated ERC20 from Giveaway contract when there are no assets or lands allocated
      ✓ User cannot claim if they claim the wrong amount of ERC20
      ✓ User cannot claim more than once
      ✓ User cannot claim from Giveaway contract if destination is not the reserved address
      ✓ User cannot claim from Giveaway contract to destination zeroAddress
      ✓ User cannot claim from Giveaway contract to destination MultiGiveaway contract address
      ✓ User cannot claim from Giveaway if ERC1155 contract address is zeroAddress (441ms)
      ✓ User cannot claim from Giveaway if ERC721 contract address is zeroAddress
      ✓ User cannot claim from Giveaway if ERC20 contract address is zeroAddress
      ✓ User cannot claim from Giveaway if ERC20 contract address array length does not match amounts array length
      ✓ User cannot claim from Giveaway if ERC1155 values array length does not match ids array length
      ✓ User cannot claim after the expiryTime (452ms)
      ✓ User cannot claim if expiryTime is 0
    Multi_Giveaway_two_giveaways
      ✓ User cannot claim when test contract holds no tokens - multiple giveaways, 1 claim (232ms)
      ✓ User cannot claim sand when contract does not hold any - multiple giveaways, 1 claim (573ms)
      ✓ User can claim allocated multiple tokens from Giveaway contract - multiple giveaways, 1 claim (1350ms)
      ✓ User can claim allocated multiple tokens from Giveaway contract - multiple giveaways, 2 claims (157ms)
      ✓ User cannot claim from Giveaway contract if the claims array length does not match merkle root array length
      ✓ User cannot claim from Giveaway contract if the claims array length does not match proofs array length
      ✓ User cannot claim allocated tokens from Giveaway contract more than once - multiple giveaways, 2 claims (65ms)
    Multi_Giveaway_single_claim
      ✓ User cannot claim when test contract holds no tokens (231ms)
      ✓ User cannot claim sand when contract does not hold any (440ms)
      ✓ User can claim allocated multiple tokens from Giveaway contract (503ms)
      ✓ Claimed Event is emitted for successful claim
      ✓ User cannot claim more than once
    Trusted_forwarder_and_meta-tx
      ✓ should fail to set the trusted forwarder if not admin (208ms)
      ✓ should succeed in setting the trusted forwarder if admin
      ✓ claim with meta-tx: user can claim from single giveaway using single claim function (499ms)
      ✓ claim with meta-tx: user cannot claim from single giveaway using single claim function more than once (59ms)
      ✓ claim with meta-tx: user can claim from single giveaway using multiple claim function (85ms)
      ✓ claim with meta-tx: user cannot claim from single giveaway using multiple claim function more than once (52ms)
      ✓ claim with meta-tx: user can claim from multiple giveaways (716ms)
      ✓ claim with meta-tx: user cannot claim from multiple giveaways more than once (81ms)

  Permit
    ✓ ERC20 Approval event is emitted when msg signer == owner (275ms)
    ✓ Nonce is incremented for each Approval
    ✓ Permit function reverts if deadline has passed
    ✓ Permit function reverts if owner is zeroAddress
    ✓ Permit function reverts if owner != msg signer
    ✓ Permit function reverts if spender is not the approved spender
    ✓ Domain separator is public
    ✓ Non-approved operators cannot transfer ERC20 until approved
    ✓ Approved operators cannot transfer more ERC20 than their allowance
    ✓ Approved operators cannot transfer more ERC20 than there is

  PolygonAsset.sol
    ✓ user sending asset to itself keep the same balance (1840ms)
    ✓ user batch sending asset to itself keep the same balance
    ✓ user batch sending in series whose total is more than its balance
    ✓ user batch sending more asset that it owns should fails
    ✓ can get the chainIndex from the tokenId
    ✓ can get the URI for an NFT
    ✓ can get the URI for a FT
    ✓ fails get the URI for an invalid tokeId
    ✓ can burn ERC1155 asset
    ✓ can burn ERC721 asset
    PolygonAsset: MetaTransactions
      ✓ can transfer by metaTx
      ✓ fails to transfer someone else token by metaTx
      ✓ can batch-transfer by metaTx (45ms)
    Asset <> PolygonAsset: Transfer
```

```
     ✓ can transfer L1 minted assets: L1 to L2 (3946ms)
     ✓ can transfer L2 minted assets: L2 to L1 (77ms)
     ✓ can transfer multiple L1 minted assets: L1 to L2 (144ms)
     ✓ can transfer partial supplies of L1 minted assets: L1 to L2 (165ms)
     ✓ can transfer multiple L2 minted assets: L2 to L1 (748ms)
     ✓ can transfer partial supplies of L2 minted assets: L2 to L1 (216ms)
     ✓ can transfer assets from multiple L1 minted batches: L1 to L2 (211ms)
     ✓ can transfer assets from multiple L2 minted batches: L2 to L1 (243ms)
     ✓ can return L1 minted assets: L1 to L2 to L1 (105ms)
     ✓ can return L2 minted assets: L2 to L1 to L2 (109ms)
     ✓ Deposit 1 asset from 20 ERC1155 L1 to L2 (72ms)
    Transfer Gems and catalyst L1 to L2
       ✓ Deposit asset from L1 to L2 with 1 catalyst legendary and 4 power gems (84ms)
       ✓ Deposit asset from L1 to L2 with 1 catalyst legendary (80ms)
       ✓ Deposit asset from L1 to L2 with 1 catalyst legendary 1 gem defense (78ms)
       ✓ Deposit asset from L1 to L2 with catalyst or gems out of bound  (46ms)
       ✓ Deposit asset from L1 to L2 without catalyst and gems (74ms)
    Transfer Gems and catalyst L2 to L1
       ✓ Deposit asset from L2 to L1 with 1 catalyst legendary and 4 power gems (94ms)
       ✓ Deposit asset from L2 to L1 with 1 catalyst legendary (95ms)
       ✓ Deposit asset from L2 to L1 with 1 catalyst legendary 1 gem defense (93ms)
       ✓ Deposit asset from L2 to L1 without catalyst and gems (74ms)

PolygonBundleSandSale.sol
   ✓ should fail to deploy with a zero receiving wallet (64ms)
   ✓ assuming that the medianizer return the price in u$s * 1e18 and usdAmount is also in u$s * 1e18. There is no need to round up at most you loose 1e-18 u$s.
   ✓ onERC1155Received should fail if called directly
  setReceivingWallet
     ✓ should fail to setReceivingWallet if not admin
     ✓ should fail if address is zero
     ✓ admin should success to setReceivingWallet
  create Sale
     ✓ NFT can't be used with numPacks > 1 ?
     ✓ NFT can be used with numPacks == 1
     ✓ single sale
     ✓ multiple/batch sale (52ms)
  Withdraw
     ✓ withdraw (132ms)
     ✓ should fail to withdraw if not admin (59ms)
  Buy packs with DAI
     ✓ should fail with the wrong saleId
     ✓ should fail if not enough packs (74ms)
     ✓ should fail if not enough DAI (74ms)
     ✓ buy with DAI, obs: buyer != to (128ms)
    Buy packs with ETH
       ✓ should fail with the wrong saleId
       ✓ should fail if not enough packs (69ms)
       ✓ should fail if not enough ETH (77ms)
       ✓ buy with ETH, obs: buyer != to (130ms)

AssetAttributesRegistry
   ✓ getRecord for non existing assetId (1275ms)
   ✓ setCatalyst for legendary catalyst with 4 gems (61ms)
   ✓ setCatalyst should fail for non minter account
   ✓ setCatalyst with gems.length > MAX_NUM_GEMS should fail (40ms)
   ✓ setCatalyst with gems.length > maxGemForCatalyst should fail
   ✓ setCatalystWithBlockNumber should fail for non migration contract
   ✓ addGems to rareCatalystId (71ms)
   ✓ should fail for non-nft
   ✓ addGems should fail for non minter account (52ms)
   ✓ addGems should fail for empty gemsId array
   ✓ addGems should fail for non existing catalystId
   ✓ should fail for gemId = 0
   ✓ addGems should fail when trying to add two gems in total to commonCatalyst (69ms)
   ✓ admin can change attributes contract
   ✓ fails if anyone other than admin trys to change attributes

AssetAttributesRegistry: getAttributes
  getAttributes: minting
     ✓ can get attributes for 1 gem (2744ms)
     ✓ can get attributes for 2 identical gems (524ms)
     ✓ can get attributes for 3 identical gems (185ms)
     ✓ can get attributes for 4 identical gems (130ms)
     ✓ can get attributes for 2 different gems (112ms)
     ✓ can get attributes for 3 different gems (122ms)
     ✓ can get attributes for 4 different gems (122ms)
     ✓ can get attributes for 2 identical gems + 1 different gem (119ms)
     ✓ can get attributes for 3 identical gems + 1 different gem (126ms)
     ✓ can get attributes for 2 identical gems + 2 different identical gems (120ms)
  getAttributes: upgrading
     ✓ can get attributes when adding 1 gem to an asset with an empty catalyst (136ms)
     ✓ can get attributes when adding 2 identical gems to an asset with an empty catalyst (139ms)
     ✓ can get attributes when adding 3 identical gems to an asset with an empty catalyst (143ms)
     ✓ can get attributes when adding 4 identical gems to an asset with an empty catalyst (144ms)
     ✓ can get attributes when adding 2 different gems to an asset with an empty catalyst (138ms)
     ✓ can get attributes when adding 3 different gems to an asset with an empty catalyst (139ms)
     ✓ can get attributes when adding 4 different gems to an asset with an empty catalyst (150ms)
     ✓ can get attributes when adding 1 similar gem to an asset with existing gems (143ms)
     ✓ can get attributes when adding 1 different gem to an asset with existing gems (141ms)
     ✓ can get attributes when adding 2 similar gems to an asset with existing gems (137ms)
     ✓ can get attributes when adding 2 different gems to an asset with existing gems (145ms)
     ✓ can get attributes when adding 3 similar gems to an asset with existing gems (144ms)
     ✓ can get attributes when adding 3 different gems to an asset with existing gems (142ms)
     ✓ can get attributes when adding gems to an asset multiple times (163ms)
     ✓ can get attributes when upgrading an asset multiple times (355ms)
     ✓ attributes after multiple upgrades are correct (168ms)
     ✓ should fail if numGems > MAX-NUM_GEMS (138ms)

AssetMinter
  AssetMinter: Mint
     ✓ the assetMInterAdmin is set correctly (1754ms)
     ✓ the assetMinter quantities are set correctly (68ms)
     ✓ Record is created with correct data on minting with legendary catalyst (NFT) (2628ms)
     ✓ Transfer event is emitted on minting an NFT (catalyst legendary) (272ms)
     ✓ CatalystApplied event is emitted on minting an NFT with a catalyst (363ms)
     ✓ Catalysts and gems totalSuplies are reduced when added (334ms)
     ✓ Mint without catalyst (342ms)
     ✓ Mint custom number admin (45ms)
     ✓ Mint custom number user3 (58ms)
    AssetMinter: MintMultiple
       ✓ TransferBatch event is emitted on minting a single FT via mintMultiple (300ms)
       ✓ TransferBatch event is emitted on minting a multiple FTs (207ms)
       ✓ CatalystApplied event is emitted for each NFT minted with a catalyst (454ms)
       ✓ records should be updated correctly for each asset minted (138ms)
       ✓ totalSupply & balance should be reduced for burnt gems & catalysts (354ms)
    AssetMinter: addGems
       ✓ Can extract an erc721 & add Gems (2684ms)
    AssetMinter: Failures
       ✓ should fail if "to" == address(0)
       ✓ should fail if "from" != _msgSender()
       ✓ should fail if gem == Gem(0)
       ✓ should fail if gemIds.length > MAX_NUM_GEMS (55ms)
       ✓ should fail if gemIds.length > maxGems (39ms)
       ✓ minting WO catalyst: should fail if asstID = 0
       ✓ custom minting: should fail if qty = 0
       ✓ custom minting: should fail if not admin
       ✓ mintMultiple should fail if assets.length == 0
       ✓ mintMultiple should fail if catalystsQuantities == 0
       ✓ mintMultiple should fail if gemsQuantities == 0
       ✓ mintMultiple should fail if trying to add too many gems
       ✓ mintMultiple: should fail if gemsId = 6
       ✓ mintMultiple: should fail if catalystsId = 5
       ✓ mintMultiple: should not set catalyst if catalystId == 0 (322ms)

AssetUpgrader
   ✓ extractAndSetCatalyst for FT with rareCatalyst and powerGem, no ownership change (1775ms)
   ✓ extractAndSetCatalyst should fail for NFT (50ms)
   ✓ setting a rareCatalyst with powerGem and defenseGem (119ms)
   ✓ adding powerGem and defenseGem to a rareCatalyst with no gems (137ms)
   ✓ setting a rareCatalyst where ownerOf(assetId)!= msg.sender should fail (82ms)
   ✓ burns sand fees when feeRecipient = BURN_ADDRESS (119ms)

CollectionCatalystMigrations
   ✓ migrating assetId with epic catalyst and no gems (2070ms)
   ✓ migrating assetId with rare catalyst and power gem (136ms)
   ✓ migrating assetId of quantity = 1 with legendary catalyst (123ms)
   ✓ migrating asset with collection id != 0 should fail (117ms)
   ✓ migrating assetId not from admin account should fail
   ✓ migrating assetId that does not exist in old registry should fail
   ✓ migrating assetId that has already been migrated should fail (179ms)
   ✓ batch migrating assetId not from admin account should fail
   ✓ batchMigrate two assets (205ms)
   ✓ setAssetAttributesRegistryMigrationContract first assignment
   ✓ setAssetAttributesRegistryMigrationContract first assignment should fail for non admin
   ✓ setAssetAttributesRegistryMigrationContract second assignment
   ✓ setAssetAttributesRegistryMigrationContract second assignment should fail for non migrationContract

GemsCatalystsRegistry
   ✓ getMaxGems for commonCatalyst should be 1 (1264ms)
   ✓ getMaxGems for non existing catalystId should fail
   ✓ burnCatalyst should burn 2 common catalysts from catalystOwner account
   ✓ burnCatalyst should burn 2 common catalysts from superOperator account
   ✓ Allow max value allowance for every gems and catalyst (38ms)
   ✓ Allow 0 allowance for every gems and catalyst (53ms)
   ✓ burnCatalyst should fail for unauthorized account
   ✓ burnCatalyst should fail for non existing catalystId
```

```
     ✓ burnCatalyst should fail for insufficient amount
     ✓ burnCatalyst should fail for account with no gems
     ✓ burnGem should burn 3 power gems from gemOwner account
     ✓ burnGem should burn 3 power gems from superOperator account
     ✓ burnGem should fail for unauthorized account
     ✓ burnGem should fail for non existing gemId
     ✓ burnGem should fail for insufficient amount
     ✓ burnGem should fail for account with no gems
     ✓ addGemsAndCatalysts should fail for existing gemId
     ✓ addGemsAndCatalysts should fail for existing catalystd
     ✓ addGemsAndCatalysts should add gemExample
     ✓ addGemsAndCatalysts should add catalystExample
     ✓ addGemsAndCatalysts should fail for gem id not in order
     ✓ addGemsAndCatalysts should fail for unauthorized user
     ✓ addGemsAndCatalysts should fail if too many G&C (51ms)
     ✓ addGemsAndCatalysts pass if max -1 G&C (55ms)
     ✓ burnDifferentGems for two different gem tokens
     ✓ burnDifferentCatalysts for two different catalyst tokens
     ✓ batchBurnGems for two different gem tokens and two different amounts
     ✓ Change trusted forwarder

  MockLandWithMint.sol
     ✓ creation (170ms)
     ✓ cannot set polygon Land Tunnel to zero address (1042ms)
     ✓ supported interfaces
    Mint and transfer full quad
      With approval
         ✓ transfers quads of all sizes (1434ms)
      Without approval
         ✓ reverts transfers of quads (731ms)
      From self
         ✓ transfers of quads of all sizes from self (2113ms)
    Burn and transfer full quad
       ✓ should revert transfer quad from zero address
       ✓ should revert transfer quad to zero address
      With approval
         ✓ should not transfer a burned 1x1 quad
         ✓ should not transfer burned quads (6588ms)
      From self
         ✓ should not transfer a burned 1x1 quad
         ✓ should not transfer burned quads (8203ms)
    mint and check URIs
       ✓ mint and check URI 1
       ✓ mint and check URI 3
       ✓ mint and check URI 6 (45ms)
       ✓ mint and check URI 12 (128ms)
       ✓ mint and check URI 24 (472ms)
       ✓ reverts check URI for non existing token
    Mint and transfer a smaller quad
       ✓ transferring a 1X1 quad from a 3x3
       ✓ transferring a 1X1 quad from a 12x12 (141ms)
       ✓ transferring a 3X3 quad from a 6x6 (59ms)
       ✓ transferring a 6X6 quad from a 12x12 (165ms)
    Mint and transfer all its smaller quads
       ✓ transferring all 1X1 quad from a 3x3 (167ms)
       ✓ transferring all 1X1 quad from a 6x6 (582ms)
       ✓ transferring all 1X1 quad from a 12x12 (2310ms)
    transfer batch
       ✓ transfers batch of quads of different sizes (1495ms)
       ✓ transfers batch of quads of different sizes from self (1223ms)
       ✓ reverts transfers batch of quads to address zero
       ✓ reverts transfers batch of quads from address zero
       ✓ reverts transfers batch of quads for invalid parameters
    Testing transferFrom
       ✓ Transfer 1x1 without approval
       ✓ Transfer 1x1 with approval
    testing batchTransferFrom
       ✓ Mint 12x12 and transfer all internals 1x1s from it (334ms)
    Meta transactions
      transferQuad without approval
         ✓ should not transfer quads of any size (2690ms)
      transferQuad with approval
         ✓ should transfer quads of any size (2161ms)
      transferQuad from self
         ✓ should transfer quads of any size (2161ms)
      Burn and transfer full quad
         ✓ should revert transfer of 1x1 quad after burn (59ms)
         ✓ should revert transfer of any size quad after burn (12132ms)
      batchTransferQuad
         ✓ should batch transfer 1x1 quads (92ms)
         ✓ should batch transfer quads of different sizes (593ms)
    Getters
       ✓ returns the width of the grid
       ✓ returns the height of the grid
       ✓ should fetch x and y values of given quad id (1492ms)
       ✓ cannot fetch x and y values of given non existing quad id
       ✓ should fetch owner of given quad id (1528ms)

  PolygonLand.sol
    Land <> PolygonLand: Transfer
      L1 to L2
         ✓ only owner can pause tunnels
         ✓ only owner can unpause tunnels
         ✓ set Max Limit on L1
         ✓ cannot set Max Limit on L1 if not owner
         ✓ set Max Allowed Quads
         ✓ cannot Max Allowed Quads if not owner
         ✓ should not be able to transfer Land when paused (258ms)
         ✓ should be able to transfer 1x1 Land (44ms)
         ✓ should be able to transfer 3x3 Land (54ms)
         ✓ should be able to transfer 6x6 Land (103ms)
         ✓ should be able to transfer 12x12 Land (273ms)
         ✓ should be able to transfer 24x24 Land (989ms)
         ✓ should should be able to transfer multiple lands (138ms)
        Through meta transaction
           ✓ should be able to transfer 1x1 Land (47ms)
           ✓ should be able to transfer 3x3 Land (62ms)
           ✓ should be able to transfer 6x6 Land (109ms)
           ✓ should be able to transfer 12x12 Land (308ms)
           ✓ should should be able to transfer multiple lands meta (153ms)
      L2 to L1
         ✓ only owner can pause tunnels
         ✓ only owner can unpause tunnels
DUMMY CHECKPOINT. moving on...
         ✓ should not be able to transfer Land when paused (95ms)
DUMMY CHECKPOINT. moving on...
         ✓ should be able to transfer 1x1 Land (82ms)
DUMMY CHECKPOINT. moving on...
         ✓ should be able to transfer 12x12 Land (514ms)
         ✓ should not be able to transfer 2, 12x12 Land at once (569ms)
DUMMY CHECKPOINT. moving on...
         ✓ should be able to transfer 3x3 Land (108ms)
DUMMY CHECKPOINT. moving on...
         ✓ should be able to transfer 6x6 Land (189ms)
DUMMY CHECKPOINT. moving on...
         ✓ should should be able to transfer multiple lands (247ms)
         ✓ should not be able to transfer if exceeds limit (86ms)
      Through meta Tx
DUMMY CHECKPOINT. moving on...
         ✓ should be able to transfer 1x1 Land (82ms)
DUMMY CHECKPOINT. moving on...
         ✓ should be able to transfer 3x3 Land (111ms)
DUMMY CHECKPOINT. moving on...
         ✓ should be able to transfer 6x6 Land (202ms)
DUMMY CHECKPOINT. moving on...
         ✓ should be able to transfer 12x12 Land (533ms)

  PolygonLandWeightedSANDRewardPool
     ✓ last time reward application should match the duration (943ms)
     ✓ total supply is at first empty
     ✓ staking should update the reward balance, supply and staking token balance
     ✓ withdraw should update the reward balance, supply and staking token
     ✓ reward per token should be 0 if total supply is 0
     ✓ reward per token calculation
     ✓ earned calculation
     ✓ get reward should transfer the reward and emit an event (44ms)
     ✓ exiting should withdraw and transfer the reward
     ✓ pool contains reward tokens
     ✓ user can earn reward tokens if pool has been notified of reward
     ✓ admin can notify to start a new reward process (without sending more reward tokens)
     ✓ user cannot earn rewardTokens if they stake after the end time
     ✓ user earns full reward amount if there is only one staker after 1 day(s)
     ✓ user earns full reward amount if there is only one staker after 27 day(s)
     ✓ User with 0 LAND earns correct reward amount
     ✓ User with 0 LAND earns correct reward amount - smaller stake
     ✓ User with 1 LAND(s) earns correct reward amount (52ms)
     ✓ User with 3 LAND(s) earns correct reward amount (71ms)
     ✓ User with 10 LAND(s) earns correct reward amount (129ms)
     ✓ User can withdraw some stakeTokens after several amounts have been staked
     ✓ First user can claim their reward - no NFTs
     ✓ First user can claim their reward - has NFTs (125ms)
     ✓ A user can claim their reward after multiple stakes (182ms)
     ✓ First user can exit the pool (39ms)
     ✓ A user can exit the pool after multiple stakes (58ms)
     ✓ A user with NFTs can exit the pool after multiple stakes (211ms)
     ✓ Change externals contracts
    ✓ user earnings for 0 NFT(s) match expected reward with 1 stake(s)
    ✓ user earnings for 0 NFT(s) match expected reward with 2 stake(s)
    ✓ user earnings for 0 NFT(s) match expected reward with 4 stake(s) (56ms)
    ✓ user earnings for 0 NFT(s) match expected reward with 10 stake(s) (116ms)
    ✓ user earnings for 1 NFT(s) match expected reward with 1 stake(s) (48ms)
    ✓ user earnings for 1 NFT(s) match expected reward with 10 stake(s) (240ms)
    ✓ user earnings for 2 NFT(s) match expected reward with 1 stake(s) (58ms)
    ✓ user earnings for 3 NFT(s) match expected reward with 1 stake(s) (64ms)
    ✓ user earnings for 3 NFT(s) match expected reward with 10 stake(s) (288ms)
    ✓ user earnings for 89 NFT(s) match expected reward with 1 stake(s) (704ms)
    ✓ user earnings for 89 NFT(s) match expected reward with 10 stake(s) (968ms)
    ✓ Multiple Users' earnings for 0 NFTs match expected reward: 2 users, 10 stake each (209ms)
    ✓ Multiple Users' earnings for 0 NFTs match expected reward: 3 users, 1 stake each (54ms)
    ✓ Multiple Users' earnings for 1 NFTs match expected reward: 2 users, 1 stake each (79ms)
    ✓ Multiple Users' earnings for 1 NFTs match expected reward: 2 users, 10 stake each (76ms)
    ✓ Multiple Users' earnings for 3 NFTs match expected reward: 2 users, 1 stake each (117ms)
    ✓ Multiple Users' earnings for 100 NFTs match expected reward: 2 users, 1 stake each (1566ms)
    ✓ Staking with STAKE_AMOUNT plus an extra amount equivalent to 2 NFTs
    ✓ Earlier staker gets more rewards with same NFT amount - small NFT number (74ms)
    ✓ Earlier staker gets more rewards with same NFT amount - large NFT number (1573ms)
    ✓ More lands give more rewards than earlier staker when NFT amounts are smaller (91ms)
    ✓ More lands do not give more rewards than earlier staker with large NFT amounts (1619ms)
    ✓ rewardToken in pool is more than amount notified
    ✓ rewardToken in pool is zero
    ✓ rewardToken in pool is less than amount notified
    ✓ the call to notifyRewardAmount is made after users first call stake
    ✓ user is earning rewards and pool is notified for a second time before end of current reward period
    ✓ Multiplier & reward are correct (111ms)
    ✓ Only sender or reward distribution can compute sender's account

  PolygonSANDRewardPool
    ✓ last time reward application should match 30 days (488ms)
    ✓ total supply is at first empty
    ✓ staking should update the reward balance, supply and staking token balance
    ✓ withdraw should update the reward balance, supply and staking token
    ✓ reward per token should be 0 if total supply is 0
    ✓ reward per token calculation
    ✓ earned calculation
    ✓ get reward should transfer the reward and emit an event
    ✓ exiting should withdraw and transfer the reward

  PolygonSand.sol Meta TX
    transfer
      ✓ without metatx
      ✓ with metatx
    approve and transferFrom
      ✓ without metatx
      ✓ with metatx (39ms)
    burn
      ✓ without metatx
      ✓ with metatx
    trusted forwarder
      ✓ should fail to set the trusted forwarder if not owner
      ✓ should success to set the trusted forwarder if owner
    approveAndCall
      ✓ without metatx
      ✓ with metatx
    paidCall
      ✓ without metatx
      ✓ with metatx

  PolygonSand.sol
    Bridging: L1 <> L2
      ✓ should update the child chain manager
      ✓ should fail if not owner when updating the child chain manager
      ✓ should fail when updating the child chain manager to address(0)
      ✓ should be able to transfer SAND: L1 to L2 (1197ms)
      ✓ should be able to transfer SAND: L2 to L1 (1185ms)
    Getters
      ✓ gets the correct name of the Sand Token
      ✓ gets the correct symbol of the Sand Token

  SandBaseToken.sol
    Deployment
      ✓ total supply should be 3,000,000,000 * 10^18 (204ms)
    Transfers
      ✓ users should be able to transfer some of the token they have
      ✓ users should be able to transfer all token they have
      ✓ users should not be able to transfer more token than they have
      ✓ users should not be able to transfer token they dont have
      ✓ users balance should not move if they transfer token to themselves
      ✓ total supply should not be affected by transfers
    Allowance
      ✓ user should not be able to transfer more token than their allowance permit
      ✓ user should be able to transfer some of the amount permitted by their allowance
      ✓ user should be able to transfer all tokens that their allowance permit
      ✓ burn test

  PolygonSandClaim
    ✓ fetches the given amount of fake sand from user and transfers the same amount of new sand (442ms)
    ✓ reverts if claim amount is more than balance
    ✓ returns the amount of sand which has been claimed

  SandPolygonDepositor
    ✓ Locking funds in sand predicate mock contract (329ms)

  RaffleTheDoggies
    - should be able to mint with valid signature
    - should be able to mint 10_000 different tokens
    - should be able to mint 10_000 different tokens in 3 waves
    - should be able to mint 10_000 different tokens in 3 waves in 3 txs

  ERC20BasicApproveExtension
    ✓ ApproveAndCall calling buyLandWithSand (1299ms)
    ✓ ApproveAndCall should fail for input data too short
    ✓ ApproveAndCall should fail for first parameter != sender
    ✓ ApproveAndCall should fail for zero data
    ✓ ApproveAndCall should fail for Approving the zeroAddress
    ✓ ApproveAndCall should work for target = EOA with ether value = 1
    ✓ ApproveAndCall should work for target = EOA with ether value = 0
    ✓ ApproveAndCall for an empty contract as a target should revert
    ✓ ApproveAndCall calling logOnCall of a mock contract
    ✓ ApproveAndCall with only one parameter should fail
    ✓ ApproveAndCall calling revertOnCall of a mock contract should fail
    ✓ PaidCall calling buyLandWithSand (217ms)
    ✓ PaidCall should fail for input data too short
    ✓ PaidCall should fail for first parameter != sender
    ✓ PaidCall should fail for zero data
    ✓ PaidCall should fail for Approving the zeroAddress
    ✓ PaidCall should work for target = EOA with ether value = 1
    ✓ PaidCall should work for target = EOA with ether value = 0
    ✓ PaidCall for an empty contract as a target should revert
    ✓ PaidCall calling logOnCall of a mock contract
    ✓ PaidCall calling revertOnCall of a mock contract should fail

  Gems & Catalysts permit
    ✓ user can use permit function to approve Gems via signature
    ✓ user can use permit function to approve Catalysts via signature
    ✓ updates a users allowances correctly
    ✓ should fail if deadline < block.timestamp
    ✓ should fail if recoveredAddress == address(0) || recoveredAddress != owner
    ✓ should fail if owner == address(0) || spender == address(0)

  Sand.sol
    Deployment
      ✓ total supply should be 3,000,000,000 * 10^18
    Transfers
      ✓ users should be able to transfer some of the token they have
      ✓ users should be able to transfer all token they have
      ✓ users should not be able to transfer more token than they have
      ✓ users should not be able to transfer token they dont have
      ✓ users balance should not move if they transfer token to themselves
      ✓ total supply should not be affected by transfers
    Allowance
      ✓ user should not be able to transfer more token than their allowance permit
      ✓ user should be able to transfer some of the amount permitted by their allowance
      ✓ user should be able to transfer all tokens that their allowance permit
    Getters
      ✓ gets the correct name of the Sand Token
      ✓ gets the correct symbol of the Sand Token

  Batch.sol coverage
    ✓ atomicBatchWithETH (200ms)
    ✓ nonAtomicBatchWithETH
    ✓ atomicBatch
    ✓ nonAtomicBatch
    ✓ singleTargetAtomicBatchWithETH
    ✓ singleTargetNonAtomicBatchWithETH
    ✓ singleTargetAtomicBatch
    ✓ singleTargetNonAtomicBatch
    ✓ onERC1155Received
    ✓ onERC1155BatchReceived
    ✓ onERC721Received
    ✓ supportsInterface
```

```
1259 passing (4m)
5 pending
```

Code Coverage

Coverage for most of the code is acceptable, except for `RequirementsRules.sol` and `ERC2771HandlerV2.sol` (coverage for both of these files is quite low). We recommend adding additional tests to get coverage close to 100%.

Fixes review (4): Coverage is still low for `ERC2771HandlerV2.sol`. Consider adding additional tests to get coverage as close to 100% as possible.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
Game/	100	82.86	100	100	
GameBaseToken.sol	100	83.33	100	100	
GameMinter.sol	100	75	100	100	
GameV1.sol	100	100	100	100	
Sand/	100	100	100	100	
SandBaseToken.sol	100	100	100	100	
Utils/	100	50	100	100	
Batch.sol	100	50	100	100	
asset/	87.46	67.84	84.09	87.44	
AssetAttributesRegistry.sol	97.33	83.33	100	97.47	170,171
AssetMinter.sol	96	80.77	91.67	96.1	216,218,334
AssetSignedAuctionWithAuth.sol	81.58	71.74	73.33	81.58	… 330,424,429
AssetUpgrader.sol	93.02	65.38	84.62	93.02	219,221,229
AssetV2.sol	93.75	50	100	93.75	45
ERC1155ERC721.sol	83.28	62.73	80	83.12	… 795,907,908
asset/libraries/	100	75	100	100	
AssetHelper.sol	100	70	100	100	
ERC1155ERC721Helper.sol	100	100	100	100	
bundleSandSale/	100	67.39	100	100	
PolygonBundleSandSale.sol	100	67.39	100	100	
catalyst/	99.06	95.45	97.06	99.06	
Catalyst.sol	100	100	100	100	
CollectionCatalystMigrations.sol	100	91.67	100	100	
DefaultAttributes.sol	100	100	100	100	
Gem.sol	100	100	100	100	
GemsCatalystsRegistry.sol	98.31	96.15	95.65	98.31	252
catalyst/interfaces/	100	100	100	100	
ICollectionCatalystMigrations.sol	100	100	100	100	
IGemsCatalystsRegistry.sol	100	100	100	100	
IOldCatalystRegistry.sol	100	100	100	100	
claims/AssetGiveaway/	96.67	93.75	90.91	96.67	
AssetGiveaway.sol	91.67	87.5	80	91.67	72
ClaimERC1155.sol	100	100	100	100	
claims/MultiGiveaway/	97.96	96.43	94.44	97.96	
ClaimERC1155ERC721ERC20.sol	100	92.86	100	100	
MultiGiveaway.sol	96.3	100	91.67	96.3	131
claims/signedGiveaway/	96.67	92.86	91.67	96.67	
SignedERC20Giveaway.sol	**96.67**	**92.86**	**91.67**	**96.67**	**133**
common/Base/	100	100	100	100	

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
TheSandbox712.sol	100	100	100	100	
common/BaseWithStorage/	83.62	63.73	84.85	83.87	
ERC2771Handler.sol	62.5	50	80	66.67	36,37,39
ERC2771HandlerV2.sol	60	37.5	80	63.64	37,38,39,41
ERC721BaseToken.sol	83.96	67.65	86.21	84.55	… 382,384,399
ImmutableERC721.sol	96	66.67	90	96	80
MetaTransactionReceiver.sol	40	0	33.33	40	14,15,27
WithAdmin.sol	100	75	100	100	
WithMinter.sol	100	100	100	100	
WithPermit.sol	100	100	100	100	
WithSuperOperators.sol	100	50	100	100	
WithUpgrader.sol	75	0	66.67	60	15,16
common/BaseWithStorage/ERC20/	93.85	71.05	90.91	93.85	
ERC20BaseToken.sol	93.55	71.05	89.47	93.55	120,148,149,150
ERC20Token.sol	100	100	100	100	
common/BaseWithStorage/ERC20/extensions/	100	50	100	100	
ERC20BasicApproveExtension.sol	100	50	100	100	
ERC20Internal.sol	100	100	100	100	
common/BaseWithStorage/ERC677/extensions/	100	100	100	100	
ERC677Extension.sol	100	100	100	100	
common/Libraries/	82.22	52.63	93.33	81.32	
BytesUtil.sol	75	50	100	80	13
ObjectLib32.sol	95.24	80	100	95	52
PriceUtil.sol	62.5	50	100	57.14	18,21,25
SafeMathWithRequire.sol	100	50	100	100	
SigUtil.sol	45	28.57	66.67	45.45	… 43,46,47,49
Verify.sol	100	100	100	100	
common/interfaces/	100	100	100	100	
ERC1271.sol	100	100	100	100	
ERC1271Constants.sol	100	100	100	100	
ERC1654.sol	100	100	100	100	
ERC1654Constants.sol	100	100	100	100	
IAssetAttributesRegistry.sol	100	100	100	100	
IAssetMinter.sol	100	100	100	100	
IAssetToken.sol	100	100	100	100	
IAssetUpgrader.sol	100	100	100	100	
IAttributes.sol	100	100	100	100	
IAuthValidator.sol	100	100	100	100	
IERC1155.sol	100	100	100	100	
IERC1155TokenReceiver.sol	100	100	100	100	
IERC165.sol	100	100	100	100	
IERC20.sol	100	100	100	100	
IERC20Extended.sol	100	100	100	100	
IERC677.sol	100	100	100	100	
IERC677Receiver.sol	**100**	**100**	**100**	**100**	
IERC721.sol	100	100	100	100	

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
IERC721Events.sol	100	100	100	100	
IERC721Extended.sol	100	100	100	100	
IERC721MandatoryTokenReceiver.sol	100	100	100	100	
IERC721TokenReceiver.sol	100	100	100	100	
IGameMinter.sol	100	100	100	100	
IGameToken.sol	100	100	100	100	
ILandToken.sol	100	100	100	100	
IPolygonLand.sol	100	100	100	100	
Medianizer.sol	100	100	100	100	
defi/	92.86	76.6	89.61	92.13	
ERC20RewardPool.sol	91.43	75	86.84	90.74	… 220,257,446
SandRewardPool.sol	93.68	81.82	91.67	92.86	… 172,235,385
StakeTokenWrapper.sol	100	50	100	100	
defi/contributionCalculation/	95.24	80	90	95.24	
LandContributionCalculator.sol	92.31	83.33	83.33	92.31	36
LandOwnerContributionCalculator.sol	100	75	100	100	
defi/interfaces/	100	100	100	100	
IContributionCalculator.sol	100	100	100	100	
IContributionRules.sol	100	100	100	100	
IRewardCalculator.sol	100	100	100	100	
defi/rewardCalculation/	80.68	83.33	84.62	80.68	
PeriodicRewardCalculator.sol	100	100	100	100	
TwoPeriodsRewardCalculator.sol	72.58	76.67	76.47	72.58	… 176,177,178
defi/rules/	91.34	74.55	84	91.74	
ContributionRules.sol	93.98	75	84.21	94.19	… 263,269,276
LockRules.sol	64.71	76.92	70	67.57	… 136,137,139
RequirementsRules.sol	97.37	72.73	90.48	97.48	186,196,373
faucet/	95.45	75	100	95.45	
Faucet.sol	95.45	75	100	95.45	89
permit/	100	100	100	100	
Permit.sol	100	100	100	100	
polygon/LiquidityMining/	91.38	63.64	93.18	91.67	
IRewardDistributionRecipient.sol	100	75	100	100	
PolygonLandWeightedSANDRewardPool.sol	95.59	70.83	95.83	95.65	226,230,232
PolygonSANDRewardPool.sol	84.44	50	88.24	84.78	… 145,149,151
polygon/child/	100	100	100	100	
ChildGameTokenV1.sol	100	100	100	100	
polygon/child/asset/	95.24	62.5	100	95.24	
PolygonAssetV2.sol	95.24	62.5	100	95.24	46
polygon/child/land/	87.83	68.89	93.33	88.93	
PolygonLandBaseToken.sol	87.01	68.45	100	87.97	… 565,569,570
PolygonLandTunnel.sol	90.7	75	82.35	92.68	97,131,140
PolygonLandV1.sol	100	75	100	100	
polygon/child/sand/	95.24	70	90.91	95.24	
PolygonSand.sol	**91.67**	**66.67**	**85.71**	**91.67**	**55**
PolygonSandClaim.sol	100	75	100	100	

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
polygon/child/sand/interfaces/	100	100	100	100	
IPolygonSand.sol	100	100	100	100	
polygon/root/	100	100	100	100	
IRootChainManager.sol	100	100	100	100	
SandPolygonDepositor.sol	100	100	100	100	
polygon/root/land/	86.96	50	76.92	86.96	
LandTunnel.sol	85	50	72.73	85	34,70,97
MockLandTunnel.sol	100	100	100	100	
raffle/	0	0	0	0	
Raffle.sol	0	0	0	0	… 217,221,225
All files	**88.85**	**69.96**	**87.68**	**88.99**	

Appendix

File Signatures

The following are the SHA-256 hashes of the reviewed files. A file with a different SHA-256 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different SHA-256 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review.

Contracts

```
a914216d177942799d7053910fbd8ff50dd11728e033f552f32613e6b175e648 ./solc_0.8/Sand/SandBaseToken.sol
8252cfd8e478c5c85070f1c6d285dae1baf481ac4651e2b188ea36861996943b ./solc_0.8/claims/AssetGiveaway/AssetGiveaway.sol
4a741df624e9683103191f5180744f6a78cfc3d8594d236852d1f399290b4cb4 ./solc_0.8/claims/AssetGiveaway/ClaimERC1155.sol
18dfe30d0e1fb63b21924a0ea0476d4e57ed21389be7e8af2f93cdd7e6e1c668 ./solc_0.8/claims/signedGiveaway/SignedERC20Giveaway.sol
fac00a5da4b30228e7beab256e2db8b7f85f94dfcb1d124ab9c4c6e66bea46c6 ./solc_0.8/claims/MultiGiveaway/ClaimERC1155ERC721ERC20.sol
8827da526afb8c372c7457a82431076eee3d74da0e479c248393a5fa020effca ./solc_0.8/claims/MultiGiveaway/MultiGiveaway.sol
7084a8ec54ede538389e68c20c61eb8e2e8bc98f240b2cdf2c1c1494e2c25931 ./solc_0.8/defi/SandRewardPool.sol
02a34ac60108610c5e88ee41d8e38854c672d32335a1fd1cb9a6fb5f55cfec54 ./solc_0.8/defi/ERC20RewardPool.sol
c80d10d43993235fbb29183725ed1ece9f9c7122bd7759b16281cc06b90ce1e6 ./solc_0.8/defi/StakeTokenWrapper.sol
80ca4b8186c0d4145250a178ecaa643ba78140e214afd339709540835f0af522 ./solc_0.8/defi/interfaces/IRewardCalculator.sol
a2d01da3a3c1b510077d429a2477453bc410bafdcbd6dfa9d15eb6c639c96d6f ./solc_0.8/defi/interfaces/IContributionRules.sol
b7b57fe1ed5d168314d13f611173caf6f5305f67a40e4e82d7e47fe4b8d84ca5 ./solc_0.8/defi/interfaces/IContributionCalculator.sol
5a4ab74b9f172d2bcf354329d405c62e4f607fecbd26a2b30bc38abb3f31b92e ./solc_0.8/defi/contributionCalculation/LandOwnerContributionCalculator.sol
52042a3227fe416286e3c379567cc87e6a089a8ed0dc1faa5cf9e2a43a4a0f8e ./solc_0.8/defi/contributionCalculation/LandContributionCalculator.sol
d2b84a1a8fe8061cc85d96eae14c63dc57830605cb0e40dd97dbda136f6d15d0 ./solc_0.8/defi/rules/RequirementsRules.sol
7b405ab53498da709b6059f464f64b43449c2472d510959f6a0674d27fadcb11 ./solc_0.8/defi/rules/LockRules.sol
6bf29112a9d242549556d95ba2fd85a0ac4281bc31537643bc4e9da66cef5f04 ./solc_0.8/defi/rules/ContributionRules.sol
0b441fcc205c4f508aa6d945527590cf7296e781eadbcf04219bdf9bb4221796 ./solc_0.8/defi/rewardCalculation/TwoPeriodsRewardCalculator.sol
2029cd331946db657679933dc1f18aca5293a007a94e976ce97508ae842d77e9 ./solc_0.8/defi/rewardCalculation/PeriodicRewardCalculator.sol
11f96884b82fac4f7a9c13434812c111eb44b88a35761272fcd29682ecf93653 ./solc_0.8/bundleSandSale/PolygonBundleSandSale.sol
0c3b264840547b2f9220b5ba4d0d1063a84e45ea0f5371b794a56b37758f5b55 ./solc_0.8/raffle/Raffle.sol
db75d5264a49f1eaf98141e05aab0903cd2db4662f9d6aae919b4b9beb33f776 ./solc_0.8/faucet/Faucet.sol
1feb56c69d0bd537e7dc10a2b17ffda69f7c76f4ae1d59b403f27928e3de505f ./solc_0.8/common/interfaces/IERC20.sol
349a855f29e62c1139028c5b839e409a32898ad5a9fc113334cde9cd6254e6b7 ./solc_0.8/common/interfaces/IERC20Extended.sol
f4f6a953ff19b4e1b7e1339c4b343bb3acf5c10b613c3ff02122f86c8e923ac9 ./solc_0.8/common/interfaces/IERC1155.sol
48fc8e152b180eb6938f45ffc4e95d921c6c663fdcbf79182a893c7f5c79443c ./solc_0.8/common/interfaces/Medianizer.sol
89a09d8771ee00d80e3654392fc032e841f25bb078882f18e010e28b58b58391 ./solc_0.8/common/interfaces/IERC721.sol
3dfa73f5ccd56261584768124d7a971944a148ad55f036b2dba37a86d54171d0 ./solc_0.8/common/interfaces/IERC1155TokenReceiver.sol
17e70e17da269b7a86e2014347a978263ed0f14c8d8273335147d4ce974913b3 ./solc_0.8/common/interfaces/IERC677Receiver.sol
61c95f6921163aad1aaf3fa42032e90f0d98d4a5ed21c646f3a28ac536dc0764 ./solc_0.8/common/interfaces/IGameMinter.sol
c08c3abb66a9fd9ac973c37a4d9220fd305115e01143c07141d8efef1b5b8c5b ./solc_0.8/common/interfaces/ERC1654.sol
e4e1b7c66fed1aaa9058c4f7a2ec69d2e761614b3be96969d1b70463075525b3 ./solc_0.8/common/interfaces/IERC721Extended.sol
c6990617481aa2f9749c0f2e772e9d6b6cf81b683624d239afded976e6d19596 ./solc_0.8/common/interfaces/IERC721Events.sol
6cda29531bee813879c504c366b34b007499d41efbed7c515dfa4c3e9a43b46d ./solc_0.8/common/interfaces/IERC721TokenReceiver.sol
beb0b0672fc99f1417613c832a79beb11e3283399bba338b93b38d66bfe58514 ./solc_0.8/common/interfaces/ERC1271.sol
d3f07d2b5224c1342fbc8f761a93e97856e41e7da26c3ff066d0b76c86ac8143 ./solc_0.8/common/interfaces/ERC1271Constants.sol
55478359342de00baab95a1e34b8bf7c1a8ff1664a84491289461bc51f97fdcb ./solc_0.8/common/interfaces/IERC677.sol
17bd397358d1dfedc2a5204ff52d5ee4962dad0113b67ae6eed486354353ae67 ./solc_0.8/common/interfaces/IAssetUpgrader.sol
```

```
cfafb021c4ac9c2e024d6164c946e42f25a30c4ab96972b865c53c424ddec6b2 ./solc_0.8/common/interfaces/IPolygonLand.sol
2bd3ce8011577e3271b7597d42ec0efb21e3c40df887d60607acebf04c658e33 ./solc_0.8/common/interfaces/IAttributes.sol
b682b2b1ba0cd860c179ea820ab87c26799704ff24b1a3e6d5ed9165f3cb1f6c ./solc_0.8/common/interfaces/IAssetAttributesRegistry.sol
9d1b706e70c51eb93b262017a861c53046978247312ac9f9b525a08495d8575e ./solc_0.8/common/interfaces/IAssetToken.sol
9073b3da579f093123bf2419fef081f9aad801b46b6170d52dc49e0752fce800 ./solc_0.8/common/interfaces/IERC721MandatoryTokenReceiver.sol
5ecdc2c6ae9e0ac11632d7981b9565bb99623874f372ac9711aab8058bc56c89 ./solc_0.8/common/interfaces/ERC1654Constants.sol
aae5dcb417dec176726e5ade31275f71f5105ba336b8497ae7b9e7b88072c11f ./solc_0.8/common/interfaces/ILandToken.sol
9b6a515fb6167a77e3ec8e084dc8922cc77f8fa9e22312281015e9993d8c6a56 ./solc_0.8/common/interfaces/IAssetMinter.sol
b61348573f905a1edbb4c3faaf1c5133bc79745302c4c6054051e2f71cabac1f ./solc_0.8/common/interfaces/IERC165.sol
cd0e4bc707a91b8c7b7604ae02c1968e51862e7fcb216b20a97e1bc2a3ec85ea ./solc_0.8/common/interfaces/IAuthValidator.sol
d576d560048afabd7766beb8b4e2f347399655046518b65b5e43d0192a0f414c ./solc_0.8/common/interfaces/IGameToken.sol
22ae6ee22b6a40d3b1b42498651447e82382ad076a3840759653ac8084aedb6c ./solc_0.8/common/Base/TheSandbox712.sol
3abdf8ec1dceaa0fa32c4a8bc2dc2060e3baf032db472bd2d8fba949bb629591 ./solc_0.8/common/Libraries/SafeMathWithRequire.sol
c8c1e1aab762ad1614a5182f4ac9a862974e9164ba7f3aec997ce77ae34a3e70 ./solc_0.8/common/Libraries/SigUtil.sol
b1d5f37f7987be434859ff101bbaa2cc34b09d25249c64dace4d85d2fa302738 ./solc_0.8/common/Libraries/PriceUtil.sol
864d8d4f6e09fcfe17025b9fc875e237838ce041de88f177f8947755c19526f8 ./solc_0.8/common/Libraries/ObjectLib32.sol
70c4007df7c51d7d29f72d2c3090d2e5c62ebc2ca8d4983842fe90ac4583a5f8 ./solc_0.8/common/Libraries/BytesUtil.sol
2b442dea7a5238a842c5532a3ca662537fb469a0a66820b9c913262e9787ea9d ./solc_0.8/common/Libraries/Verify.sol
020b381a262857b77a2c079a308c5ba1af16a12f8a0b1e8ed4a41dd27e5c9efa ./solc_0.8/common/BaseWithStorage/WithAdmin.sol
70e28124291b77fbcff215a4fea8eb0a0726ed444ae2234284aa71227f55d171 ./solc_0.8/common/BaseWithStorage/ERC721BaseToken.sol
c7672ef2210c6788720fc2cdbb36912564500de9b332155de9fc45818edf4dab ./solc_0.8/common/BaseWithStorage/WithSuperOperators.sol
6cade310be0f69cb2e331995e678e5b07ff96958173a25f6dfb37a7e48dc4316 ./solc_0.8/common/BaseWithStorage/WithUpgrader.sol
d0ee27182854aedb2973f004f6d634b005c48cf75a4dbb7199ca48c470445a51 ./solc_0.8/common/BaseWithStorage/ImmutableERC721.sol
7add3196afdd12ae1ddeca90d7c7be4c3650949fe58b750f7d269d2fb7ec2362 ./solc_0.8/common/BaseWithStorage/ERC2771Handler.sol
a1c4c89c876230b3de886b483ea4fd32649f645ac569f7213fc3a67e461c2db8 ./solc_0.8/common/BaseWithStorage/ERC2771HandlerV2.sol
d5da80553c31d72af39eeeef5424b54651889506cc8facdf212d0c6da8419149 ./solc_0.8/common/BaseWithStorage/WithPermit.sol
e5d639d4b8dd7a77865111d054bb0cde545f5577b4443afa3ef8d65d49fb5ea1 ./solc_0.8/common/BaseWithStorage/MetaTransactionReceiver.sol
5abee99b49456dedf0a6705876f135a59a816d2ba82508306c37be458258cc43 ./solc_0.8/common/BaseWithStorage/WithMinter.sol
715bb97bdf093c2a034983a16d9fb8dae9501f4ccf20fdae431e2b99ac945b37 ./solc_0.8/common/BaseWithStorage/ERC677/extensions/ERC677Extension.sol
264cdd6dbd4cfad633bbde440f356a2faa45d79cca70acf760c27a9237b6010d ./solc_0.8/common/BaseWithStorage/ERC20/ERC20BaseToken.sol
95b045ca14ba2025a5f2d678c3d884847b4d53b6807f55cfbec12e7a6210bb8a ./solc_0.8/common/BaseWithStorage/ERC20/ERC20Token.sol
242ec99cc12aaf649e702a2889977e53901a5df2168a835f611c7dba6be431f2
./solc_0.8/common/BaseWithStorage/ERC20/extensions/ERC20BasicApproveExtension.sol
086140896c8bc0b525aa47fa100ae5796693a4a2eeb333b51260022cda9f358b ./solc_0.8/common/BaseWithStorage/ERC20/extensions/ERC20Internal.sol
b594b414e5043158f7a4182ac7428c075e221ad982ac355c3786397291fe6a2a ./solc_0.8/permit/Permit.sol
1339aa61c5c0adeb9101cecc338e1d3f76da2dc376d908dc8b518a9e6514e035 ./solc_0.8/Game/GameBaseToken.sol
35451a0cfa92bb1963e6bab516ed8875dffe413bbd45f8b99ab08a5f733bfa17 ./solc_0.8/Game/GameV1.sol
89e91ef71043ee2cafda621c79158e651628d561493621fc1324694c6f5e96c8 ./solc_0.8/Game/GameMinter.sol
9248ea04b181a95ff8ece3301b8057c4dd6095add17ca34a86ad34e705e11235 ./solc_0.8/Utils/Batch.sol
093813bee5f84475b5722929e2f738b8887318e1a4f8c547cef1c79dcd55a746 ./solc_0.8/polygon/child/ChildGameTokenV1.sol
d4254210527f4e0f5365e7e9a5d8d810ee646f4f08e96e7daa379f651a71c01a ./solc_0.8/polygon/child/sand/PolygonSandClaim.sol
db278a0f7b057936efe440d16b777df06ec5351b771ecf670f0230db7b7d25a7 ./solc_0.8/polygon/child/sand/PolygonSand.sol
8a069f5dcc240fa42645712f1782ad48b54360cce96e70b3dca9e40e5d0b6fb9 ./solc_0.8/polygon/child/sand/interfaces/IPolygonSand.sol
43f7af82a986b145c9e5c4199494a44dbd520129f814f6d8dd35b7fcfcb2e9d8 ./solc_0.8/polygon/child/asset/PolygonAssetV2.sol
6c4d8a9017f4708c0dbcca0c794c131e57df0a7a61783ef54427a2766f207b3d ./solc_0.8/polygon/child/land/PolygonLandBaseToken.sol
9b424f87d48a620c4b06b08ace25ce3a37f992095c8d61f73674fd5ca151a3d6 ./solc_0.8/polygon/child/land/PolygonLandTunnel.sol
df7bb1183a51a729822e8309fc123194e9bb12b1d600d6f81b52e5788f619262 ./solc_0.8/polygon/child/land/PolygonLandV1.sol
c8dcaff076f21c4d8052f5ae415afc19f09abbe490bb1c80d35abd05e7705da3 ./solc_0.8/polygon/root/SandPolygonDepositor.sol
bda48918996ed8e556c1d58eedea6d8c717a2658214bca51a2397c7e508d78da ./solc_0.8/polygon/root/IRootChainManager.sol
78fe5432325d8e4010ad4c0b2c91a6764f1f4a776473cdc825b05d25b82fa7e4 ./solc_0.8/polygon/root/land/LandTunnel.sol
64b04161a7b321a2637075e75ce6b472e0ce9bf0641b0c77196a8bc19ee68956 ./solc_0.8/polygon/root/land/MockLandTunnel.sol
63e3fab77b323891ba35b40598bfadf4a71c0a3af6610a8759a926ad18dd90be ./solc_0.8/polygon/LiquidityMining/PolygonSANDRewardPool.sol
d9e98831b5c0012bd7fcf37dfc42632149d6348fff104d31715e7c6c9647358c ./solc_0.8/polygon/LiquidityMining/IRewardDistributionRecipient.sol
ea40786181d290d5e23d075f6fe4af27022af95cf341b7818a0f482824f9a3f1 ./solc_0.8/polygon/LiquidityMining/PolygonLandWeightedSANDRewardPool.sol
ed9c660f72d0dfdca454ee89c0fc2ac6baaa36d1b3f9feb059634dc0d22386ea ./solc_0.8/catalyst/DefaultAttributes.sol
b698d23bef1cd9c2740e0daa9a7a4c696216dc05949930ce9937c4ad173b3c87 ./solc_0.8/catalyst/GemsCatalystsRegistry.sol
9aa83b5f007967e081fbd40598321a5ab864c80e26d0a5ea42282a4f81a94d2d ./solc_0.8/catalyst/Catalyst.sol
6f6ca6cc6e150dfbda3baae712c6ad07fb908ca59d365ad2e3ede6b06a3cceaa ./solc_0.8/catalyst/CollectionCatalystMigrations.sol
4b3673afad8ca2b9f0e31a2e19c6500335869407dee4fd5a51374c041cea56e0 ./solc_0.8/catalyst/Gem.sol
786f1b121358bb6a6c070752db6976a6821f51916e7f887f9c34902be6487dcd ./solc_0.8/catalyst/interfaces/IOldCatalystRegistry.sol
155012cb55ffa9ee9eb5809a4a4dd5dd395e3e512ab3a5790bac92bf7dbb5041 ./solc_0.8/catalyst/interfaces/ICollectionCatalystMigrations.sol
714e429d511faad4dd4e34c86e3dd0abef4203a058d51f1022daa420733ef7b2 ./solc_0.8/catalyst/interfaces/IGemsCatalystsRegistry.sol
```

```
17dff3853659cfdd033d8e2f491f48863bfdd70152c79f794af58a043320c71b ./solc_0.8/test/FakePredicateForwarder.sol
21c6fe73fd6e5a4ac4f79107fe68f3c409bf0b58e8e8abf77521cd6b113ada58 ./solc_0.8/test/MockERC667Reciever.sol
4df0b956011788a79115b2763bf9ccfbaaa268701d4dd08e792e0ed399bcfff4 ./solc_0.8/test/PolygonLandWeightedSANDRewardPoolNFTTest.sol
efd1cc02debe98cf58f0f352d17c18e2503c5fb974c7407c6f2f8ebca7d98efb ./solc_0.8/test/FakeChildChainManager.sol
02aab072e5aed2e3631b5cae93fd680f583860ebcdd01e1cbf682f5b04f466f4 ./solc_0.8/test/ERC721Mintable.sol
f44a14feacefa57976832224b20e3c8c94d9149442d9cf91ce603394f68c709a ./solc_0.8/test/FakeLPSandMatic.sol
2d0fb99c1b433e98821ee5fd51bc836307e0a42da54940d4478ee8e65b7a2d8f ./solc_0.8/test/EmptyContract.sol
c9930a288d0f5ed718a8714fe868e94b7ae5afb05a776866f2ac6beaca60cd8d ./solc_0.8/test/ERC1155Mintable.sol
0c0322879c113504f2f85fa54427d00d75da97b24bc07266ebcf543d4639f6f5 ./solc_0.8/test/MockLandWithMint.sol
d4d3e915d101dc52cce86d2cfb65f8c92bb45733b5a1ac54ab87ec1b4605dd32 ./solc_0.8/test/PayableMock.sol
09ac59f30be69b23cbf1a951eef6848610ae8ce3669b2436c837f7daa41df4dc ./solc_0.8/test/RewardCalculatorMock.sol
8eb0207621a57290e20dba905adf50463f4c5cd4930f2b78d2f074c8901739be ./solc_0.8/test/FakePolygonLand.sol
ec8e9512d58020464023f6eef890b477f50a8d5f1ff8c3826138bb8be47e9abe ./solc_0.8/test/FallbackContract.sol
c2b73d9a5e63e2ccf34b74a77ea0fc96d73677da2d09f63c6ac53d569934cee9 ./solc_0.8/test/ContributionCalculatorMock.sol
54a6b6450ebd70c77c31c1a52c08459afa8953b47088458297b9ddc60c485919 ./solc_0.8/test/MockERC20BasicApprovalTarget.sol
fd0ef49d6c3d4783c2035b1009dd033a3ceee5f75ca35442ce76a3287cb88597 ./solc_0.8/test/MockAssetAttributesRegistry.sol
1ee4b2d585fbb777ba874810776d5164f0f1749edf22ffac513467736049012f ./solc_0.8/test/FakeFxRoot.sol
eabfd0f77dc78950629f9eb063dc2690fe72e64814dd9ba0b58d65542a18057e ./solc_0.8/test/TestMetaTxForwarder.sol
c935ecb5fc4890ad5e16688d5273b1c7da114339b2f52420673ca39ed17a84fa ./solc_0.8/test/FakePolygonSand.sol
789d7a70718a400de1e9a8fbfb3ac9630b2cba36af477825efce94c280007a93 ./solc_0.8/test/MockSafeMathWithRequire.sol
fc43be0c148497f632f6bda31169da45a83f128f85bab28526f33751be148cfa ./solc_0.8/test/FakeERC20Predicate.sol
52ee0ad767c2a70d5f0b8736504bf019a7563614277a20ffd0069479b6345ad1 ./solc_0.8/test/TestEIP712.sol
db06e266c1c8d2102af759f59973ac1795ecaa01a38ce64899dbe34cfb31c543 ./solc_0.8/test/ContributionRulesMock.sol
2dbc0b141873653d3d0852cde97586462ccde65357d0c79bc71c24bdaeb2ba6e ./solc_0.8/test/FakeCheckpointManager.sol
17013f2f950122dc36be34dc19bd49450fd194e366f2470cd7c23717ca39fccc ./solc_0.8/test/FakeFxChild.sol
531f53934c58b13060ec928461dffc568e5c9f4b5fde8f43423b7a713d6ec41b ./solc_0.8/test/ERC20Mintable.sol
1bf5cdd3293509192b758ca16b009d39b35b1c5f0107c30190de266e040233e8 ./solc_0.8/test/AssetUpgraderFeeBurner.sol
f8cac11c3f9a2786e2e32c5983d01f505378026dc73a23548d83a911f71a8b27 ./solc_0.8/test/FakeERC1155Predicate.sol
9ae7f4ac8028192ed585efb64a922ae799d657b35f41289231743f5958346ff4 ./solc_0.8/asset/AssetV2.sol
9f6618de2653023e5270918e235a3b36af232cd6aa465b48eb1ee9135a2de4a9 ./solc_0.8/asset/AssetMinter.sol
18b690c9cbedfa17aac501b44c9cf0cd2e0f5fcac42483ac94f5accdb00dc870 ./solc_0.8/asset/AssetUpgrader.sol
1171946b36869c87cb521101a6995b26123e72e02e0c2ca385eec3d66434e188 ./solc_0.8/asset/ERC1155ERC721.sol
3b2972468f156a715761d6a5b39d8bce46b7e79f081194d8c0890cf4a4c0a01c ./solc_0.8/asset/AssetSignedAuctionWithAuth.sol
489d8d27daa728fe71c9d8594759574f0103571afe25a96e9253b5ca25dfbca8 ./solc_0.8/asset/AssetAttributesRegistry.sol
f399f35c347642fab505326a19f1634018122a9dcbe65aec8d8b4e6bfd2d00a9 ./solc_0.8/asset/libraries/ERC1155ERC721Helper.sol
49376abc7978f62fd1e72953fbe75d5356be09fa6473db4be04c5edaa32befa1 ./solc_0.8/asset/libraries/AssetHelper.sol
```

Tests

```
9e32f60dfdfc031ca589cf789f48458a023533a1d87d38894471f134123e513c ./test/sendMetaTx.ts
4e2585a6ddb2147ba235c09d16feee3b1e16264ccbfbf04b7ebf9a17b248c841 ./test/chai-setup.ts
d31d98b375dbef571645e45b48ba571a0215b73e51681365f61ea13cfe527414 ./test/utils.ts
c1b5c8c0197083d7216bab7d946557171356e5546f90b26209eec31240c888be ./test/forwardRequestData712.ts
202266a6189fcff67bcc7ce9e59a463de535fa4d4a6798e6a1b16b75ca2c6ad8 ./test/chaiBigNumberCloseTo.ts
ce685453a3762463b449bfb92619a55245b0f683e2379e9e4796f23ebc3bc8f3 ./test/evmSnapshotRevert.test.ts
7beff9569c7bd158ae5fad2a58fb0d2608764a2ff23d427ff36bd5c10b254a98 ./test/sand/fixtures.ts
2b61b76c0976d7920ec16807d6dd2d2d8ad49b44c731caeca1192e112f93d8bb ./test/sand/erc20BasicApproveExtension.test.ts
52ead9788e626791cf23b33d5753d2e2c17858b1a471d66d2f3dd9a15d407325 ./test/sand/ERC20Permit.test.ts
a1de6ff93ae1a906d4ab84863a371deda13fc277c275f6bd28b204aa52d11833 ./test/sand/Sand.test.ts
bdff7e20506d8e2c47e387f1e13aa75fb32bf0bd194ca43818d8a8e82cdcd081 ./test/erc1155/index.js
ee022b3c676541fc4662d00f93c7b29d842c5cd73035301aa20a1ecfa676ce7d ./test/claim/signedGiveaway/fixtures.ts
ca8447e666f25ec322cb764b4a36e580066bc242bf8485dfe2754031604d6f9e ./test/claim/signedGiveaway/signature.ts
c83ef66b3c535dbb007057ab4fc3bb49fcdd9224af66e89d7f8597e4ca44c1e0 ./test/claim/signedGiveaway/signedGiveaway.ts
9d6e3c49ceea136e876dd2d26cc8ff451d0201cade1d758e26d2e35ac7430e53 ./test/defi/sandRewardPool/sandRewardPool.test.ts
c2e31c1e43465663db1f22325878f30a02da46d382118321d08ec9c24474a17e ./test/defi/sandRewardPool/utils.ts
c38b3efc3aeab6823ba606c2d4e6631baf11db822d617c072d960e40986d81f4 ./test/defi/sandRewardPool/originalSet.test.ts
f4fb9c8cc47ec789b9731ed8e8b870d53b8da01870cb7aa8bdf86133632ea2e2 ./test/defi/sandRewardPool/sandRewardPool.e2e.test.ts
082b4da1ad32bfefffeee59d97849a0603dacb38c1dd90727ab2877c91e630c ./test/defi/sandRewardPool/sandRewardPoolAntiCompund.test.ts
ab61d477415ac2c7ccb4950bc62b885d01acece29213aae4e2f834b84e62b4ec ./test/defi/sandRewardPool/fixtures/contributionCalculator.fixture.ts
f71066f11fd8a404d783d5e7513facd61df4b01a562ff7f6eb2335d37311c320 ./test/defi/sandRewardPool/fixtures/rewardCalculator.fixture.ts
87717154cdbe0797a6f3a88c154365cbcf845b2716b12249914cbc5cfff0212c ./test/defi/sandRewardPool/fixtures/sandRewardPool.fixture.ts
aa7353b2c49a3dcffad0a9ebdf4e4a0f61aaba07a46a9ddc0120d84427108edf ./test/defi/sandRewardPool/fixtures/originalFixtures.ts
```

```
19dc982906fd0afb2f5060eac9756aaf9494c9c46f88a17898038b18b908655e
./test/defi/sandRewardPool/calculators/landOwnerContributionCalculator.test.ts

dce71f8644f7a3b55a3764b895b5627b0ece6c6a494abac9e734aa68533d563d ./test/defi/sandRewardPool/calculators/landContributionCalculator.test.ts

8e3c3f3a043f3bf957fdaa43a04a85eda8ced5b42f0bc04f870c2af7fe3a1ef1 ./test/defi/sandRewardPool/calculators/twoPeriodsRewardCalculator.test.ts

2b209c3326eb06987dc7e22d5453547ac11bd3138a5609d4d5dbe31a15237625 ./test/defi/sandRewardPool/calculators/periodicRewardCalculator.test.ts

aa127eea500c5a07f9ddb4287fb3f07a5e15ab41b0ff9e403fe98b54fbcd133d ./test/defi/erc20RewardPool/erc20RewardPool.test.ts

8b2e3797d44f0d8b5eb71a3db4619a3db41121240c7be6ee4daf4764d1dcaa1e ./test/defi/erc20RewardPool/fixtures/fixtures.ts

fcfbfc4b0374d61f3cb931851367f5c3c8ff74cba3c7fe35f0cf5e292132888b ./test/defi/erc20RewardPool/rules/LockRules.test.ts

4b6cc9e7b2ae910f95e4961c19961b0f9a43899a710314630267d0cff1c96d8b ./test/defi/erc20RewardPool/rules/RequirementsRules.test.ts

67cf0fdc41c5b1302597a2c4cb90db54499f5259d9139e1b8eea30689df82e49 ./test/defi/erc20RewardPool/rules/ContributionRules.test.ts

b2a2b836d163d6fe29d53a48ad8f81b2f268e4c9c1d42b0964272767fdf19089 ./test/land-sale/fixtures.ts

0f6f777865707c6088160cfc875d59fc649324ad21154adf63f255647f993c57 ./test/land-sale/EstateSaleWithAuth.test.ts

4ef0f64fe9b9c4f95f81ca4aefc80e68ed8251abe92f4a86b72c122feaa3385b ./test/land-sale/AuthValidator.test.ts

99819c1d28bdb6c2bc782825126d365105a12f62846730fe81ea80cc9f1a5580 ./test/erc721/index.js

129366cff63ba13e162d8f09a03c617f1db23cc6710f173f73672d632b15acee ./test/raffle/raffle.test.ts

d519a8acc9cb619ed1ee4bd47fc7845a37453010bd73f43b0eeab135e857aa5e ./test/raffle/fixtures.ts

32039eae645393e0f28ec66841a3c3f8edb6b58dcda64f0feb804485837fee11 ./test/assetGiveaway/fixtures.ts

0e631c666c0ed83c69931627790dcfb40f24f98b0b21ce183377894ebb88f818 ./test/assetGiveaway/assetGiveaway.test.ts

01ee014a7bb074dc88f10485612bc362faaf994019685f2f51de63852574452a ./test/assetGiveaway/gas.test.ts

cbcb3b04e57d4056958eb618bc4d1c67ba7d6fe91bdb59561f210215d34e7f1c ./test/assetGiveaway/merkleTree.test.ts

75515bd70629a9f2533fdd11be2ae7ec5216734419f84dbbf00b6cc1f34fe122 ./test/faucet/fixtures.ts

6d00a2d48163c5a504702f421b872bd258f4294174fd9092ac180b9140ad3d0b ./test/faucet/faucet.test.ts

31756ce78a0b647e4ff5a0f494db0685020954113d63593cf61149aa5febd4f7 ./test/erc677/fixtures.ts

b23a4d22756f3e776c87be1a9fb66267a9e90bc0e7189345c8408a9f791e7764 ./test/erc677/ERC677.test.ts

f648bfccfb744fbbb2028183317782e59309d6c56f778c52f0f1fdd7ad00b261 ./test/multiGiveaway/fixtures.ts

690490ae892f516a5c2e227bdfd8e2dfcdf8f92cca52785b8eb5ae749daa3570 ./test/multiGiveaway/multiGiveaway.test.ts

d8337ff0c027f771b549306859151a60ffc86978aefe4bfa283a4c9fd3a1a26b ./test/multiGiveaway/balanceHelpers.ts

ea43bf3dcf7acd814bcc92f89cee2eec8eb9110fd954e7c6df2ee1d737ff55b5 ./test/multiGiveaway/gas.test.ts

40119487b7614d062cce7fab896f84c2b989c0f2939f855b48a3e21b9eb38811 ./test/multiGiveaway/merkleTree.test.ts

e1097fde2100ae26f0abd99b9794b9f4726831f112846c4e8101c488882d143c ./test/common/contributionEquation.ts

003546a9d771e98cb5253e9776ae5d807c254ba583e7866ee7ce2714158515c0 ./test/common/fixtures/gemAndCatalysts.ts

ce7090737921ca5fa6db7355539e8c3165dfefde7b1c88a17daed0da5f1ebbca ./test/common/fixtures/erc20BasicApproveExtension.ts

c3145d379e9c168a721a4b8c7cdd94ab227a9e7523b734e95d6452a5c41fae8b ./test/common/fixtures/assetAttributesRegistry.ts

ca36768e4126a2cf1272ba1aacf391f69e9d28037d8342edca0dc932c34dcdfd ./test/common/fixtures/asset.ts

beb6ef837233134200d6ac663cbe1d89c7317529f0b7e41c2676c1883d705d47 ./test/common/fixtures/assetUpgrader.ts

f5461b339b1849ced82083e51c9d71fd262a0de851159e5b722a4fd467a23951 ./test/common/libraries/safeMathWithRequire.ts

cb019cdadcb0720c95436db9dc00799227d1a201e7c26980b7117cc79abc1b7d ./test/permit/fixtures.ts

e4a05f8b305350caf07ec7a0fc83423ae78c58b3bb832842e1929907f8bbb140 ./test/permit/permit.test.ts

2ef7c4c68fec114c2718812fa7a3f5816ee7752072b4b61fca29895a227f8e77 ./test/permit/data712.ts

1b49885663e44edf917c834415cfc94eb858d07de746fb5f68604c940d949c7f ./test/Game/fixtures.ts

0a1f020ca8a1426dd717fbf7de6699f756f1e53d7dd35cc581083d37d9b74245 ./test/Game/testERC721.test.js

3fc6933b88240bb58690742350745be58e7d6e7315401ccd27bacceb8c9148e7 ./test/Game/GameToken.test.ts

5a10830aae480e06b6e0643c55d2b8a47cdc1c4e1a1c903c7575cbd401e65dc5 ./test/Game/assets.ts

c1b5c8c0197083d7216bab7d946557171356e5546f90b26209eec31240c888be ./test/Game/data712.ts

5330049eb83769808a63d13d52f80c19e0f3af96bf6b334ffb116234628c7146 ./test/Game/gameMinter/gameMinter.test.ts

c4c9b763c555ac9d378b788a24f93d60e08ebaf87f99bac176bc56eb35c12121 ./test/utils/batch.test.ts

fd36f25cf913392c8e5e83c06c4c90ed7face7cab3e448fb180ffd1f136f9bb6 ./test/polygon/sand/fixtures.ts

95541f43f6f5f5bc0f7132f71d97ca01000a2e5de20c04fcc00f301f645f1e2e ./test/polygon/sand/sand-metatx.test.ts

6f601c7ee566878b3d863cb19d5e07305295e1fa1d6aa97dbf1df926135be067 ./test/polygon/sand/sand.test.ts

594831422c7334078301c453e877e2bc8f10ccd6e7c1cdc6776b364c1cd5ee94 ./test/polygon/sandPolyonDepositor/fixtures.ts

bac73da75a5899d168d1ce67cdde705127d2120bc12efc549542c8b8b8b7a640 ./test/polygon/sandPolyonDepositor/sandPolygonDepositor.test.ts

b806141f82e8d0ba332beadc5018e2f7265463ae2e6d9c5d4aff9a0d0b5fcf52 ./test/polygon/bundleSandSale/fixtures.ts

268ca8c41c8bc0a82a4b5cbcb17d29bd60c47c19a4c6230a70763aa2bf48b909 ./test/polygon/bundleSandSale/bundleSandSale.test.ts

6956ac6aeba350418c4fa37722ff5865480e6ef1ffaa23910e1f5877764240ff ./test/polygon/sandBase/sandbase.test.ts

4a91af4128d0bf0753e368d4e080650017e3c291e76d310cc79172effd650ae6 ./test/polygon/catalyst/utils.ts

e10cb32b373e7d01f4b00bd9b8854191e2ba619adb727aae293ce82de5189f09 ./test/polygon/catalyst/collectionCatalystMigrations/fixtures.ts

715c6db78c401e2e87ccb91b89afc764abd3af777441cb44c23b80b7bd58d202
./test/polygon/catalyst/collectionCatalystMigrations/collectionCatalystMigrations.test.ts

8756addece7034f0351744f294a03136400d82b1dbf5191604122ae84123e46d ./test/polygon/catalyst/assetAttributesRegistry/fixtures.ts

45f45ebebe2a1417f06bf301783219b5df34cfaa84eff59731015eb5ace9b412
./test/polygon/catalyst/assetAttributesRegistry/assetAttributesRegistry.test.ts

b9f79ac7d46caa62afa0c71ff59cab36feba1ed05cb0b0532a94526ed322329e ./test/polygon/catalyst/assetAttributesRegistry/getAttributes.test.ts

b7534e1690bdbfbb45926bc7b13f38e62844e9202376e802870c2ad1966f32c0 ./test/polygon/catalyst/assetMinter/fixtures.ts
```

```
c5d562b23d56d7b0894d273fe35b2b60920669378d511e216e66c148c8b39c52 ./test/polygon/catalyst/assetMinter/assetMinter.test.ts
e656182bc11b5ab3b9c28cc63f7550a6b81e38918a2995ad6a6bf5fc63301cdf ./test/polygon/catalyst/gemsCatalystsRegistry/catalyst.test.js
bab1f8fac31b984cdec01654ba26bc9edff1de9d56cff4fcafe1ba2af03d4a47 ./test/polygon/catalyst/gemsCatalystsRegistry/gemsCatalystsRegistry.test.ts
526dd82dbd0d9703d2c230ac011d09d3ac3bc64e60146d7be22de04166e0899c ./test/polygon/catalyst/gemsCatalystsRegistry/gems.test.js
a686099adb5d4801f4f08d716cf64c51b4c0fa08d76f8e241def386f92e0b36a ./test/polygon/catalyst/assetUpgrader/assetUpgrader.test.ts
5253799c99048bacfc5a2f82ea2043e396c04aa1f5173bc49295a3fc4bbd1341 ./test/polygon/asset/fixtures.ts
949561f53e85e19ca03fdcf788dd257f19f87810409d01cee061e502e7495c5f ./test/polygon/asset/asset.test.ts
c16f09e9c8e954db6e3cab94525262ab3a2ec6e2c5b5b62a19d33ad7ae9889b0 ./test/polygon/liquidityMining/fixtures.ts
ca83cdcb353f7df968f7f426cd1cbacd8fe72f5e7b77658ba8f3aaf4a2ecf765 ./test/polygon/liquidityMining/polygonLandWeightedSANDRewardPool.test.ts
7cae5c9423b098dbcc4f7cda8aa82b6332f7765fdd491dfdfa68b8fb4d2739b9 ./test/polygon/liquidityMining/polygonSANDRewardPool.test.ts
042216994b54f28c3110930d1a28ca5a3e88d45fdbe997d8f00f258be1f63213 ./test/polygon/sandClaim/fixtures.ts
ccf30de50eb5a833eb30840eb90f28a0486fb900dc6243cf43ede72751d49de8 ./test/polygon/sandClaim/polygonSandClaim.test.ts
cb08f8ab14fc5108a621f9382d66106712f47e73f44260f16e8264578dd20072 ./test/polygon/land/landTunnels.test.ts
0ad6d0adedd43e6e0189fe5b0456436fbbb48d559e28d2cf077bb33f10c4f207 ./test/polygon/land/fixtures.ts
7fcfb0818fb12bc1594fee46c1d80013ba9f3b676491338d01a7d16755ce7c3e ./test/polygon/land/testERC721.test.js
25e9d12d7a4ee67fa961957c9eca97f02ebf34740095151a1d082f9d7395fd18 ./test/polygon/land/land.test.ts
306e15dbbf40f2dcbd3442d9fb1825b92cfb5a4153c003533112d514ee934a8d ./test/erc20/index.js
c345a3919317d6a7a9f4e72434191e50e8d45fb5e8f3d8658b6eda161ca229d3 ./test/asset/asset.test.ts
cdd54628ab27ef887e9e3d9211e751a2b30df4b8d93a30c217ff544a98bb6907 ./test/asset/assetERC721.test.js
8f2befda78ff078531afaab2cd9af3370c6daf6941559603de57e7e83f6d9ea5 ./test/asset/assetERC1155.test.js
24491e9e72573a20c0fc9031eb62334e308a77952b5fa7ca16d82e76f72878f0 ./test/asset/assetSignedAuctionWithAuth.test.ts
5ae7b1e3b742b624b07e2c84a9fd7dba1a1815801b1cc1a98350ff7754c3603e ./test/liquidityMining/_testHelper.js
11d7ac4fb2c753e8fbadafbaa312597129dc2a2478f3ec62f18f1e9afa23733c ./test/liquidityMining/SANDRewardPool.test.js
4c1bd4e849b1d1527e8760bdd197727d53c0782f63c1ab8b4ef3106c42450f67 ./test/liquidityMining/mockSANDRewardPool.test.js
03d7855ef6e22c20254d7bf63f190365bb17873646734bf8c12038dc207e0e98 ./test/liquidityMining/contributionEquation.test.js
```

Changelog

- 2022-05-20 - Initial report
- 2022-06-28 - Re-audit report
- 2022-08-01 - Re-audit report (2)
- 2022-08-11 - Re-audit report (3)
- 2022-09-19 - Fixes review report (4)

About Quantstamp

Quantstamp is a Y Combinator-backed company that helps to secure blockchain platforms at scale using computer-aided reasoning tools, with a mission to help boost the adoption of this exponentially growing technology.

With over 1000 Google scholar citations and numerous published papers, Quantstamp's team has decades of combined experience in formal verification, static analysis, and software verification. Quantstamp has also developed a protocol to help smart contract developers and projects worldwide to perform cost-effective smart contract security scans.

To date, Quantstamp has protected $5B in digital asset risk from hackers and assisted dozens of blockchain projects globally through its white glove security assessment services. As an evangelist of the blockchain ecosystem, Quantstamp assists core infrastructure projects and leading community initiatives such as the Ethereum Community Fund to expedite the adoption of blockchain technology.

Quantstamp's collaborations with leading academic institutions such as the National University of Singapore and MIT (Massachusetts Institute of Technology) reflect our commitment to research, development, and enabling world-class blockchain security.

Timeliness of content

The content contained in the report is current as of the date appearing on the report and is subject to change without notice, unless indicated otherwise by Quantstamp; however, Quantstamp does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following publication.

Notice of confidentiality

This report, including the content, data, and underlying methodologies, are subject to the confidentiality and feedback provisions in your agreement with Quantstamp. These materials are not to be disclosed, extracted, copied, or distributed except to the extent expressly authorized by Quantstamp.

Links to other websites

You may, through hypertext or other computer links, gain access to web sites operated by persons other than Quantstamp, Inc. (Quantstamp). Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites' owners. You agree that Quantstamp are not responsible for the content or operation of such web sites, and that Quantstamp shall have no liability to you or any other person or entity for the use of third-party web sites. Except as described below, a hyperlink from this web site to another web site does not imply or mean that Quantstamp endorses the content on that web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the report. Quantstamp assumes no responsibility for the use of third-party software on the website and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any outcome generated by such software.

Disclaimer

This report is based on the scope of materials and documentation provided for a limited review at the time provided. Results may not be complete nor inclusive of all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Blockchain technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.