Response to the audit report findings

Identified issues for PxTrainerAdventure

Critical

1. Logical flaws in the ERC721 transfer token function check

We have fixed the issue in the smart contract and added unit tests to validate the changes.

Medium

1. TreasureTransferred Event does not contain the data required

We have fixed the event and added unit tests to validate the changes.

Gas optimisations

Primary gas usage

First claim	Second Claim	10 Claims in Total
195337	238939	2345788

Optimized gas usage

First claim	Second Claim	10 Claims in Total
198978	295266	1030181

Gas optimization suggestions 1,2 and 3

The first three gas optimization suggestions are based on claiming multiple treasures in a single transaction for a single user in a specific week. We agree that claiming multiple treasures in a single transaction would have saved gas usage.

But in production, a user will hardly get more than two treasures in a single week. The business team has also decided that a user can win *a **maximum** of 2 treasures* in a single week.

And if we observe the optimised gas usage after applying the 1,2,3 suggestion, we will find that it doesn't decrease gas consumption for claiming one or two treasures in a single

transaction. We would have benefitted from this optimisation if most of the users were capable of winning 5 or more treasures in a single week.

Also claiming multiple treasures in a single transaction will bring changes in the front-end and back-end.

Gas optimization suggestions 4

We also agree that keeping signature-related functionalities in a separate smart contract is not a smart move. But we were forced to do that.

Previously they were in the same smart contract. But as the business logics ware increased, our bytes code for the PxTrainerAdventure exceeded more than 24576 bytes.

And we were getting this error:

```
Warning: Contract code size is 27083 bytes and exceeds 24576 bytes (a limit
introduced in Spurious Dragon). This contract may not be deployable on
Mainnet. Consider enabling the optimizer (with a low "runs" value!), turning
off revert strings, or using libraries.
  --> contracts/RevisedPixelmonTrainerAdventure/PxTrainerAdventure.sol:22:1:
```

So, we had to divide the signature functionalities into a separate smart contract