



Galaxy Gamers

Des Xpheres, des Gold Pass, des Mythic Pass... Une collection « Lotpak » et un jeu de société à venir. Rien que ça 😊

[Voir le site Internet ☺](#)

Liens directs

Smart Contract : Polygonscan

DApp de claim : <https://claim.lotpak.com>

Approché par l'équipe de **Galaxy Gamers** et notamment **DamPunisher** (fondateur), c'est avec plaisir que j'ai accepté leur proposition de prendre en charge leur nouveau projet **Lotpak**, rejoignant ainsi une communauté très soudée et bienveillante.

De nombreuses missions ont été réalisées, allant de la **génération de la collection NFTs**, au **smart contract** associé (au fonctionnement moins conventionnel qu'à l'accoutumée) sans oublier la conception d'une **DApp** dédiée au **claim d'une version holographique**. Mon rôle fut aussi celui de **conseiller technique** pour le projet et son futur, qui aboutira, entre autre, à la sortie d'un **jeu de société**.

Conseiller technique

La collection **Lotpak** est la première collection indépendante de **Galaxy Gamers**, les précédentes étant liées à OpenSea...

Et qui dit première véritable collection de l'équipe, dit nécessité d'être aiguillé.

Mon rôle a été d'identifier très clairement les besoins (et leurs prérequis/pièges associés) et de proposer une solution adéquate.

L'objectif ici était de récompenser les holders de précédentes collections avec un « free mint » Lotpak.

Ainsi le choix de faire un smart contract **d'airdrop** (plutôt que de mint) fut proposé et le choix de la blockchain **Polygon** validé.

Cela permettait en prime de régler le problème du délai en supprimant la nécessité d'une DApp, de quoi proposer rapidement quelque chose de fonctionnel, qui était l'un des besoins exprimés.

Réalisation du Smart Contract

Cette collection a une autre particularité, celle d'obtenir une **version holographique** (physique) des **NFTs** possédés, avec un claim unique (à vie).

L'idée a été de stocker ceci directement dans le **Smart Contract**, de développer des fonctions dédiées à la lecture/écriture de ces infos et d'avoir une **URI IPFS** qui varie en fonction de l'état du claim, permettant ainsi d'avoir l'information bien visible sur les marketplaces.

De plus tout le nécessaire pour les **royalties** (vitales pour un projet « Free Mint ») a été mis en place, avec l'**ERC-2981** & le désormais plus que nécessaire **Operator Filter** d'**OpenSea**... Le tout régulé par un **splitter de paiement** afin de pouvoir dispatcher les fonds sur différents wallets de l'équipe.

L'airdrop fut ensuite lancé, chose rendue possible grâce à la blockchain sélectionnée : **Polygon**, permettant d'injecter directement la liste des destinataires dans le SC et de boucler pour procéder à l'airdrop. Ne surtout pas tenter l'expérience sur **Ethereum** 😊

```
Extrait du Smart Contract Lotpak

// [Owner + TokenOwner] Permet de claim la version holographique de son token, si le token existe, qu'aucune version n'a été claim auparavant et que l'on est bien contract owner ou token owner
function claimHolo(uint256 tokenId) external nonReentrant {
    require(!_exists(tokenId), "Inexistent token");
    require(!tokenHoloClaimed[tokenId], "Holographic version already claimed.");

    if (msg.sender != owner() && msg.sender != ownerOf(tokenId)) {
        revert("Only the token owner or contract owner can claim holographic version.");
    }

    tokenHoloClaimed[tokenId] = true;
}

// Retourne l'URI, soit dans sa version de base en mode "unrevealed", soit dans sa version définitive avec commutateur lié à l'état du claim holographique
function tokenURI(uint256 tokenId) override public view returns (string memory) {
    require(!_exists(tokenId), "Inexistent token");

    if (!revealed) { return uri; }

    string memory holoClaimedStr = tokenHoloClaimed[tokenId] ? "1" : "0";
    return string(abi.encodePacked(uri, tokenId.toString(), "_", holoClaimedStr, ".json"));
}

// [Owner] Effectue un airdrop, à condition qu'il s'agisse du premier & que le nombre de destinataires soit inférieur ou égal au supply total
function airdrop() external onlyOwner nonReentrant {
    require(!airdropAchieved, "Airdrop already achieved");
    require(recipients.length <= supply, "Not enough tokens to airdrop");

    // On boucle sur les destinataires en effectuant un mint à chaque fois, avec un ID débutant à 1
    for (uint256 i = 0; i < recipients.length; i++) { _mint(recipients[i], i + 1); }

    // Si le nombre d'adresses de destinataires est inférieur au supply, on transfère le reste des tokens vers l'adresse de backup
    if (recipients.length < supply) {
        for (uint256 i = recipients.length; i < supply; i++) {
            _mint(backupAddress, i + 1);
        }
    }

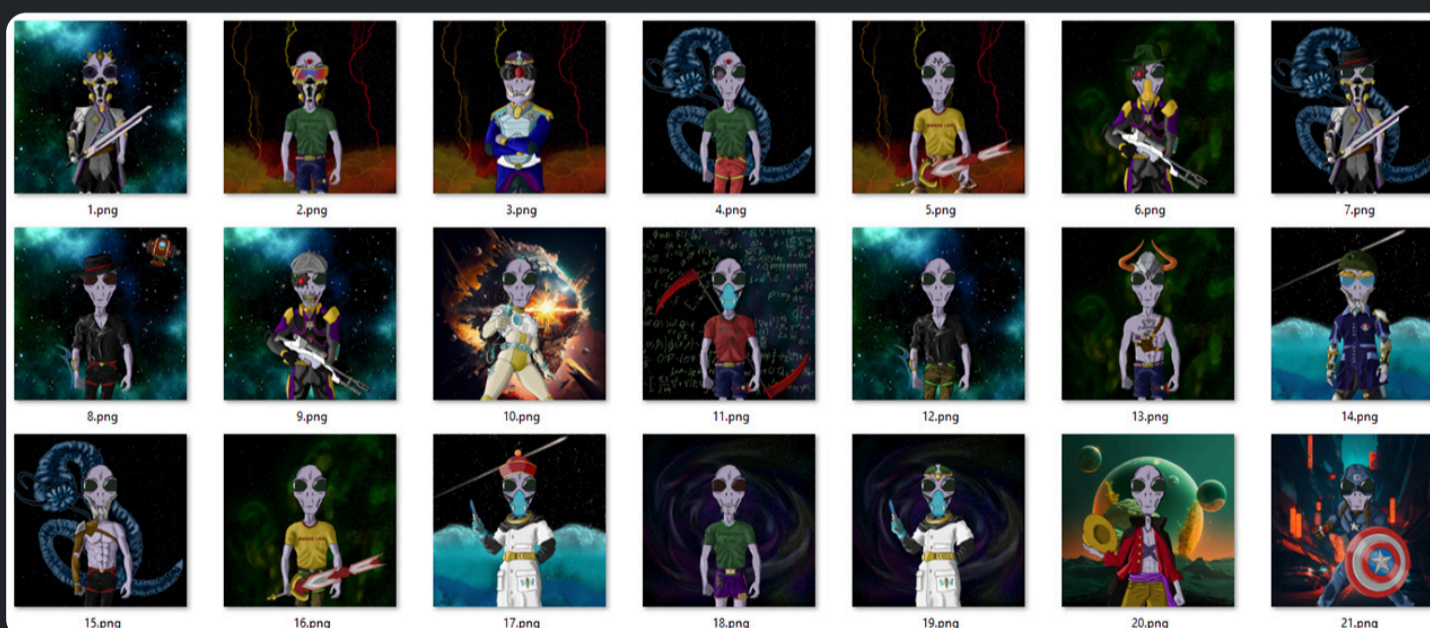
    airdropAchieved = true;
}
```

Génération des 555 NFTs

Un grand classique, avec comme toujours des **combinaisons forcées**, des **incompatibilités**, et bien entendu un **long travail de vérification**. Un travail réalisé en collaboration avec **l'artiste** derrière cette collection : **Kirk** (pour qui c'était une première) !

Cette collection avait l'originalité de proposer 3 types de **Lotpak** : **Mercenaire**, **Kug Army** et **Unique**... Avec, pour chacun, des spécificités et des contraintes particulières.

Une fois la collection validée, elle fut chargée (images et métadonnées JSON) sur le protocole **IPFS**.



Conception d'une mini DApp

Dernière brique pour ce projet « tout-en-un » : la réalisation d'une **DApp** permettant de claim la version **holographique** d'un NFT.

L'objectif était de proposer **quelque chose de très simple pour l'utilisateur final**, avec un système d'étapes menant au claim « on-chain » de sa version holographique.



0 MATIC 🟡 0x89...6011 ▾

Lotpak - Claim version Holographique

ID du Lotpak 🌱 concerné ID du Lotpak

1/3. Vérification propriétaire

Smart Contract : 0xD121FfcEc225a1aD94Ec6c6013791224890B2440
DApp made with ❤️ by RpGmAx



© 2025. Site sans cookie intrusif (aucun suivi des visites)