PRÉSENTATION DU SMART CONTRAT CALYSTIO

INTRODUCTION

Caly Token est un contrat intelligent qui implémente le standard BEP-20, permettant ainsi la création d'un jeton sur la blockchain Binance Smartchain. Ce Smart Contract est conçu pour une utilisation dans des applications décentralisées (DApps) et offre des fonctionnalités telles que les transferts de tokens, l'approbation de transfert par d'autres adresses et la brûlure de tokens.

Ce document de support fournit une explication du rôle de chaque fonctionnalité disponible dans ce Smart Contract et servira d'audit de sécurité et de fonctionnalité pour les utilisateurs qui souhaitent comprendre comment ce contrat a été conçu.

FONCTIONNALITÉS

balanceOf(address owner) public view returns (uint)

La fonction balanceOf permet de récupérer le solde de tokens d'une adresse spécifique. Cette fonction est publique et ne modifie pas l'état du contrat. Elle prend en entrée une adresse et renvoie son solde correspondant.

transfer(address to, uint value) public nonReentrant whenNotPaused returns(bool)

La fonction transfer permet de transférer les tokens depuis l'adresse de l'appelant vers une autre adresse. Les tokens sont retirés du solde de l'appelant et ajoutés au solde de l'adresse destinataire. Cette fonction vérifie si les adresses passées en entrée ne sont pas autorisées à effectuer des opérations de trading. Il vérifie également le montant à transférer avec le maximum autorisé par transaction. La fonction renvoie un booléen pour indiquer si la transaction a réussi.

transferFrom(address from, address to, uint value) public nonReentrant whenNotPaused returns(bool)

La fonction transferFrom permet de transférer les tokens depuis une adresse tierce, après approbation préalable. L'adresse émettrice doit avoir donné l'autorisation spender pour exécuter cette opération. Elle suit également les mêmes vérifications que la fonction transfer.

approve(address spender, uint value) public nonReentrant whenNotPaused returns (bool)

La fonction approve permet à une adresse donnée d'exécuter un certain nombre de transactions transferFrom. Cette fonction nécessite l'entrée de la quantité de tokens approuvés ainsi que l'adresse qui sera autorisée à effectuer des transactions en votre nom. Elle assure également les mêmes vérifications que la fonction transfer.

burn(uint256 value) public whenNotPaused returns (bool)

La fonction burn propose de bruler un certain nombre de tokens en retirant ces derniers de l'adresse émettrice et mettant à jour le paramètre totalSupply. Cette opération est irréversible.

pause() public onlyOwner

La fonction pause permet au propriétaire du contrat de mettre en pause les transactions jusqu'à nouvel ordre. Cette fonction est réservée pour le propriétaire uniquement et empêchera toute transaction ultérieure tant qu'elle n'est pas levée.

unpause() public onlyOwner

La fonction unpause permet au propriétaire du contrat de lever la pause précédemment mise en place.

addAddressToBlackList(address account) public onlyOwner

La fonction addAddressToBlackList permet d'ajouter une adresse à la "liste noire" des adresses bannies qui ne peuvent pas effectuer de transactions, cette liste est accessible uniquement pour le propriétaire du Contrat.

removeAddressFromBlackList(address account) public onlyOwner

La fonction removeAddressFromBlackList permet de supprimer une adresse de la "liste noire"

isBlacklisted(address account) public view returns (bool)

La fonction isBlacklisted renvoie un booléen pour savoir si l'adresse donnée est actuellement interdite de trader.

maxTokenPerTx(uint _maxTxAmount) external

La fonction maxTokenPerTx permet au propriétaire du contrat de fixer le maximum de tokens qui peuvent être transférés lors d'une seule transaction.

maxTokenPerWallet(uint _maxWalletAmount) external

La fonction maxTokenPerWallet permet au propriétaire du contrat de fixer le maximum de tokens qu'une adresse peut contenir.

EXPLICATION SPÉCIFIQUE LIÉE AU DIVERS FONCTIONS DU SMART CONTRAT

Token

Le contrat Token est le contrat principal qui gère la création et l'émission des jetons "CALY". Il possède les fonctions suivantes:

balanceOf(address owner): Cette fonction renvoie le solde en Token d'un propriétaire de jeton Caly

transfer(address to, uint value): Cette fonction permet à un propriétaire de jeton de transférer une quantité donnée de jetons Caly à un autre utilisateur. Elle vérifie que:

L'adresse de destination n'est pas égale à 0

L'adresse de l'envoyeur n'est pas dans la liste noire (blacklist)

L'adresse du destinataire ne figure pas dans la liste noire

La quantité de jeton transférable est inférieure ou égale à la limite maximale de transaction

Le solde du portefeuille de l'envoyeur est supérieur ou égal à la quantité de jetons à envoyer

Si toutes ces conditions sont remplies, le transfert est effectué avec succès.

- **transferFrom(address from, address to, uint value):** Cette fonction permet le transfert de jeton pour un utilisateur quelconque et surtout autorise les transferts au nom d'autres utilisateurs essentiels pour les échanges sur les échangeurs décentralisés .

Les mêmes conditions que celles appliquées à la fonction transfer s'appliquent ici.

approve(address spender, uint value): cette fonction permet aux propriétaires de tokens d'autoriser un destinataire spécifique à dépenser un certain nombre de jetons en leur nom.

burn(uint256 value): Cette fonction permet aux propriétaires de brûler un certain nombre de leurs jetons CALY. Une fois qu'une certaine quantité de jetons est brûlée, elle ne peut plus être récupérée.

Restrictions

Modifier whenNotPaused(): Cette restriction garantit que certaines fonctions ne peuvent pas être appelées par les utilisateurs lorsque les transactions sont suspendues (pausées)**.**

paused : Ce booléen indique si les transactions sont en pause ou non**. De plus l'intérêt de la présence de la fonction "paused" dans le smart contrat de Caly réside dans le faite qu'il permettra de protéger le contrat ainsi que nos utilisateurs en cas d'attaque majeur de la Blockchain. Face à ce genre de situation dès l'activation de la fonction paused tout transaction sera impossible sur le smart contrat du jeton Caly.**

onlyOwner(): Cette fonction restreint certaines actions aux seuls propriétaires de contrat (l'équipe de développeur et le CEO de Calystio)

maxTokenPerTx(uint _maxTxAmount): Cette fonction limite le montant maximal de jeton pouvant être transféré lors d'une unique transaction par adresse. **La présence de cette fonction permet à Calystio d'empêcher la vente d'une trop grosse quantité de jetons sur le marché ce qui risquerait de faire effondrer le prix et mettre en danger les investisseurs. Rassurez-vous Le montant maximum de jeton vendu ou transférer sera relativement grand et n'empêchera pas la vente des jetons.**

maxWalletAmount(uint _maxTxAmount): Cette fonction limite le montant maximal de jeton pouvant être conservé sur un portefeuille. **La présence de cette fonction permet à Calystio de garder et d'assurer une participation équitable à sa DAO, de réduire le risques de baleines (whales) et d'empêcher tout abus du smartcontrat.**

Liste noire (blacklist)

La présence d'une blacklist dans le contrat source code de Caly est justifiée. Elle est là pour anéantir le risque de bot qui surveillerait le marché pour faire des trades incessants

blacklist(address account): C'est une liste contenant les comptes qui ne sont pas autorisés à effectuer des transactions de jetons.

addAddressToBlackList(): Permet l'ajout d'un compte à la liste noire.

removeAddressFromBlackList(): Retire un compte de la liste noire.

Évènements

Les événements permettent un log clair et transparent de toutes les transactions liées à la blockchain. Les événements sont stockés sur la blockchain et peuvent être consultés ultérieurement.

Voici les événements créés dans le contrat de Calystio :

Transfer(address indexed from, address indexed to, uint value) : Emis lorsqu'un transfert de token est effectué.

Approval(address indexed owner, address indexed spender, uint value) : Emis lorsqu'une autorisation de transfert de token est accordée

Burn(address indexed burner, uint256 value) : Emis lorsqu'une quantité de Jetons Caly est brûlée.

Pause() : Emis lorsque les transactions de jeton sont en pause

Unpause() : Emis lorsque les transactions de jetons peuvent à nouveau être effectuées.

AddedToBlacklist(address indexed account) : Emis lorsqu'un compte a été ajouté à la liste noire**.**

RemovedFromBlacklist(address indexed account) : Emis lorsqu'un compte a été retiré de la liste noire.

La documentation du smart contract du jeton Caly donne une idée claire sur comment fonctionne et comment utiliser le jeton Caly.