Rapport TD ERC721

Ewan Debove

# Contrat token ERC721

J’ai basé mon contrat de définition du token sur le model Openzeppelin.

La fonction *mint* doit pouvoir servir de base pour une fonction de reproduction.

J’ai passé la fonction *\_mint* de internal à public pour pouvoir tester les contrats, elle doit être repassée en internal une fois les tests validés.

# Contrat MarketPlace

Le contrat qui sert de place de marché n’est pas un contrat ERC721. Il importe le contrat du token et contient une fonction qui permet d’instancier l’adresse du contrat du token pour pouvoir en utiliser les fonctions.

Le contrat contient une structure Sale qui sert à stocker les informations des tokens à vendre.

La structure contient : l’id du token, l’adresse du vendeur, le prix de vente, un booléen qui indique s’il s’agit d’une vente aux enchères, l’enchère la plus haute, l’adresse du plus haut enchérisseur.

Les structures Sale sont listées dans une liste.

La fonction *deposit* permet de mettre un token en vente. Le propriétaire du token doit d’abord *approve* ce contrat pour l’autoriser à manipuler le token qu’il souhaite vendre. La fonction *safeTransferFrom* le token à mettre en vente et créée un nouvel objet sale avec les informations nécessaires. L’enchère la plus haute est initialisée égale au prix, dans l’idéal il faudrait que le vendeur décide s’il veut autoriser sont token à être vendu directement et que les fonctions d’achat vérifient ces paramètres mais je n’ai pas eu le temps de travailler là-dessus.

La fonction *cancelSale* sert à retirer un token de la vente. La fonction vérifie que le msg.sender est le propriétaire du token puis supprime la sale associée, *approve* le msg.sender et lui *safeTransferFrom* son token.

La fonction *buy* permet d’acheter un token qui est en vente. Il faut que le msg.value soit au moins égal au prix affiché. S’il est supérieur l’acheteur reçoit sa monnaie. La monnaie et l’argent de la vente sont transférés à l’acheteur et au vendeur via *transfer*. La fonction approve l’acheteur et lui envoie son token via *safeTransferFrom*. La structure sale est ensuite supprimée.

La fonction *bid* sert à enchérir sur une enchère. Si la vente est bien une enchère et si le msg.value est supérieur à la dernière enchère l’ancien plus gros enchérisseur récupère son argent, le msg.sender devient le nouveau bidder et l’enchère est mise à jour.

La fonction *cashout* permet au vendeur de clore l’enchère. L’acheteur reçoit son token via *safeTranferFrom* après avoir été *approve* et le vendeur son argent via *transfer*. La vente est supprimée. Si personne n’a enchérit le prix de vente est égale à la plus haute enchère et l’acheteur ne peux pas *cashout*, autrement il pourrait recevoir des ethers gratuit en mettant un token aux enchères gratuitement et en utilisant *cashout* pour récupérer la plus haute mise (qui est initialisée égale au prix).

# Tests

J’ai essayé de mettre en place des tests pour vérifier les contrats mais je n’ai pas réussi à les faire fonctionner. Je pense que le problème vient de l’instanciation des contrats