Digital Asset Transfer via NFT (Non-Fungible Token)

Core Idea

We implement the digital asset transfer by breaking it into multiple phases where we generate the NFT using Library Implementation for our digital asset. This is followed by hosting an auction where prospective bidders can submit their bids.

After all the bids are submitted and the bidding time is complete, the auction is closed, followed by the transfer of the Ethereum from the Auction winner to the Beneficiary.

We take the input of the artist and the digital asset as a memory string and attach it to the creator of the transaction (the user who is creating the transaction). As we have a digital asset and its artist but no NFT for it, we create the NFT Token for the Digital Asset.

A Brief Outlook of the mechanism to generate the NFT Token:

- To generate/transfer an NFT for any digital asset to be stored with the miner, we use a library implementation referenced from ERC721 of the *OpenZepellin* Contract library on GitHub.
- It uses the principle of keeping a global index counter consisting of a very large unsigned integer.
- To create a new NFT token, the global index counter is incremented making it a unique and unrepeatable token.
- The incremented global index counter is then assigned to the fixed digital asset in an immutable manner meaning it cannot be edited with another value.
- It is then minted for the digital asset miner who keeps the token.
- There is no clause of decrementing the global index counter and it can only go up making it impossible to generate the same counter value for a particular asset.
- For the next asset, the counter is incremented and assigned making it impossible to revert to the original value.

The creation of the NFT Token is done using the same mechanism as stated above and we store the global index counter as our NFT Token for the digital asset given to us as input. Once the NFT is generated, we attach it to the owner/miner (creator of the transaction).

The NFT owner then starts the Auction process as the Host and sets the bidding time as deemed necessary. This bidding time is the time for which the Auction is open to bids from prospective bidders to buy the Digital Asset. It can vary from 2 hours to 2 months as the owner pleases.

To prevent useless bids and unnecessary and uninterested bidders we have kept the lowest bid of 1 ETH and all bids must comply with a value more than the lowest bid. Since there are no previous bids, the highest bid is set to ZERO and the highest bidder’s address to NULL

Once the auction is open, prospective bidders can place their bids which can be done by entering the bid value in the bidder’s contract value. As there is a bid, it must be higher than the current highest bid otherwise it is rendered meaningless.

Once a new bid is placed which is higher than the current highest, the highest bid is replaced with the value in the new bid and the highest bidder is replaced by the new bidder. As there is a submission of a bid, Ethereum is paid to the smart contract, the entry of a new highest bid will lead to the refund of the previous highest bid to the previous highest bidder.

Once the bidding period has elapsed, we close the bidding process and close the Auction with the prospective bidder who put the last valid bid (i.e., the situation when the Auction was not closed) being the Winner of the Auction. As we store only bids bigger than the previous; the last valid and prospective bid will be the highest.

Note: After the bidding period has elapsed, anybody can close the Auction as nobody can alter the result of the Auction by calling the Close Auction function.

Once, the winner has been computed, the royalty (5 % of the Highest Bid value) is transferred to the Artist's address as we need to transfer this royalty value on every transfer of the digital asset. The remaining money (95 %) is transferred from the Highest Bidder to the Digital Asset Owner.

Once, the Ethereum Transfer is complete on both ends, the ownership of the NFT is changed from the current Digital Asset Owner to the Highest Bidder. With this, we have completed the entire process of the Digital Asset Transfer from the Owner to another person via an Auction.

The new owner receives the NFT after the Ownership is transferred. If the new owner wants to resell the Digital Asset, the new owner can create another Auction by calling the same function from its address and can repeat the entire process following the same constructs stated above.

In this manner, we can resell a Digital Asset as many times as we please followed by transferring ownership and paying royalty.

Smart Contract Implementation

```
(Library Import)
Import the Library files from the ERC721 OpenZepellin Contract library on GitHub

(Smart Contract)
Start Contract

  (Library Invocation)
  Obtain the Library Global Index Counter value (Private variable in Library file)

  (Function)
  Function to Make An NFT: No function parameters : Returns NFT Token ID (Private) (Cannot let anybody generate it as it is internal)

    (Initialization)
    Initialize local variable to store the obtained Global Index Counter value
```

```
        (Updates)
        Increment the Library Global Index Counter value
        Store the current Library Global Index Counter value as an NFT Token (local variable)

        (Return)
        Return the NFT Token value (The generated NFT has to be attached with the Digital Asset)

    (Structure Definition)
    Definition of NFT Structure: No structure parameters

        (Initialization)
        Initialize memory to store the
        Owner Address,
        Artist Address,
        Digital Asset as a Memory String and
        NFT Token value

    (Initialization)
    Initialize memory of the basic variables to store the Host of the Auction, Closing Time, Top Bidder, Top Bid,
    Completion of Auction, Running of Auction, Lowest Bid, and NFT Structure

    (Constructor Definition)
    Define the Contract Constructor for the NFT to be with the Owner: Address of the Artist and Digital Asset String

        (Updates)
        Set the Owner field in the NFT structure to be the Creator of the Constructor
        Set the Digital Asset Memory field with the Input Digital Art String
        Set the Artist Address with the Input of the Artist's Address
        Set the NFT Token value with the returned Library Global Index Counter of the Make An NFT function call

    (Function)
    Function to Start the Auction: Bidding Time (public) (Will be called by the NFT Owner)

        (Basic validity conditions are checked)
        Check whether the person who has called this function is the NFT Owner (Only the NFT Owner can call this function)

        (Updates)
        Set the Address of the Top Bidder to NULL (No bid has been made)
        Set the value of the Top Bid to ZERO (No bid has been made)
        Set the Boolean Value of the Completion of Auction to False (Auction is not over)
        Set the Boolean Value of the Running of Auction to True (Auction has begun)
```

Add the Bidding Time *(in seconds)* to the Current Block Timestamp as the time remaining for the Closing Time of the Auction

Set the Host of the Auction to the Creator of the Auction

Set the value of the Lowest Bid to 1 ETH *(Lowest bid is set to the minimal positive currency value)*

(Function)

Function to put a bid: No function parameters *(public) (Will be created by a prospective bidder when the Auction is Running)*

(Basic validity conditions are checked)

Check the condition that the bid is before the Closing Time i.e., Time elapsed has not crossed the Closing Time *(Auction must not be closed)*

Check the validity of the Address of the Bidder *(The bidder cannot be invalid)*

Check that the prospective Bid is bigger than the Current Highest Bid and Lowest Bid of 1 ETH *(The highest bid is always considered, all others are disregarded)*

Check that the Auction is Running and is not Closed *(Auction must be still open to bids)*

(Updates)

Set the address of the Top Bidder to the Address of the bidder *(Replace the Highest Bidder)*

Set the value of the Top Bid to the value of the prospective bid *(Replace the Highest Bid)*

(Refund) (Ethereum Transfer)

Transfer the money of the Previous Highest Bid back to the Previous Highest Bidder *(As we have a new highest bid and bidder, the old one is not required)*

(Function)

Function to Close the Auction: No function parameters *(public) (Will be called by the NFT Owner)*

(Basic validity conditions are checked)

Check that the time elapsed has crossed the Closing Time *(Cannot close it when bids are being accepted)*

Check that the Auction has not been previously closed *(Auction must have not ended before as we can end the Auction only once)*

Check the validity of the Address of the Top Bidder *(The winning bidder cannot be invalid)*

Check the validity of the value of the Top Bid *(The winning bid cannot be invalid)*

(Updates)

Set the Boolean Value of the Completion of Auction to True *(Auction is over)*

Set the Boolean Value of the Running of Auction to False *(Auction has stopped running)*

(Ethereum Transfer)

Transfer 5 % of the money of the Highest Bid to the Artist's Address *(5 % Royalty is given to the Artist after every Auction)*

Transfer 95 % of the money of the Highest Bid to the Host of the Auction *(Digital Asset Owner) (Remaining money to the Owner)*

(Interaction)

Transfer the Ownership of the NFT Token to the Highest Bidder *(Winner of the Auction)* from the Host of the Auction

End Contract