Properties Specification of `AuctionManager`

This document specifies the invariants and function properties of the `AuctionManager` contract. Properties are written with a high level of abstraction, which allows for brevity while describing an essentially infinite number of scenarios. The focus is on having a strong collection of invariants and supplement with a minimal number of function properties.

Utilities

We will use the `tokenBalance` utility function to compute the expected amount of `_token` ERC20 tokens owned by the `AccountManager` contract.

```solidity
function tokenBalance(address _token) public view returns (uint256) {
    uint256 total;
    for (uint256 i; i < auctions.length; ++i) {
        if (bestBids[i].bidder == address(0))
            continue; // This auction has been settled
        if (auctions[i].itemToken == _token)
            total += auctions[i].amount;
        if (auctions[i].bidToken == _token)
            total += bestBids[i].amount;
    }
    return total;
}
```

The `depositedTokenAmount` function will serve to compute the amount of ERC20 tokens expected to be deposited into to `AuctionManager` by a given `_user` address.

```solidity
function depositedTokenAmount(address _token, address _user)
public view returns (uint256) {
    uint256 total;
    for (uint256 i; i < auctions.length; ++i) {
        if (bestBids[i].bidder == address(0))
            continue; // This auction has been settled
        if (auctions[i].itemToken == _token && auctions[i].seller == _user)
            total += auctions[i].amount;
        if (auctions[i].bidToken == _token && bestBids[i].bidder == _user)
            total += bestBids[i].amount;
    }
    return total;
}
```

Invariants

There are as many best bids as there are auctions,

$$\text{auctions.length} = \text{bestBids.length}.$$

All of the auction manager token amounts are accounted for (solvency),

for each address t which is the address of an ERC20 token,
$$\text{IERC20}(t).\text{balanceOf}(\text{address(this)}) = \text{tokenBalance}(t).$$

An auction can only be settled after the end time stamp,

for any $i <$ `auctions.length`, if `bestBids`$[i]$`.bidder` $=$ `address(0)` then `auctions`$[i]$`.endTime` $\leq$ `block.timestamp`.

For unsettled auction, if best bid is 0 then best bidder is the seller,

for any $i <$ `auctions.length`, if `bestBids`$[i]$`.amount` $= 0$ and `bestBids`$[i]$`.bidder` $\neq$ `address(0)` then `bestBids`$[i]$`.bidder` $=$ `auctions`$[i]$`.seller`.

For unsettled auction, the best bid is lower bounded,

for any $i <$ `auctions.length`, if `bestBids`$[i]$`.bidder` $\neq$ `address(0)` and either `bestBids`$[i]$`.bidder` $\neq$ `auctions`$[i]$`.seller` or `bestBids`$[i]$`.amount` > 0, then `bestBids`$[i]$`.amount` $\geq$ `auctions`$[i]$`.` $\geq$ `minBidAmount`.

Auction info is constant,

for any $i <$ `auctions.length`, the auction `auctions`$[i]$ is a constant.

Function Properties of `openAuction`

A non-reverting call

`openAuction(amount, itemToken, endTime, bidToken, minBidAmount)`

returns `auctions.length` $- 1$ and the auction at index `auctions.length` $- 1$ is

```
Auction({
    seller: msg.sender,
    amount: amount,
    itemToken: itemToken,
    endTime: endTime,
    bidToken: bidToken,
    minBidAmount: minBidAmount
})
```

and the best bid at index `auctions.length` $- 1$ is

```
Bid({
    bidder: msg.sender,
    amount: 0
})
```

Function Properties of `auctionBid`

Before calling `auctionBid`, let `prevBid` $=$ `bestBids[auctionId]`.

A non-reverting call `auctionBid(auctionId, amount)` will update the state by

- Updating the best bid,

$$\text{bestBids}[\text{auctionId}].\text{bidder} = \text{msg.sender} \text{ and}$$
$$\text{bestBids}[\text{auctionId}].\text{amount} = \text{amount}$$

- If $\text{prevBid.bidder} \neq \text{bestBids}[\text{auctionId}].\text{bidder}$ then

$$\text{auctions}[\text{auctionId}].\text{bidToken.balanceOf}(\text{prevBid.bidder})$$

  is increased by prevBid.amount.

- If $\text{prevBid.bidder} = \text{bestBids}[\text{auctionId}].\text{bidder}$ then

$$\text{auctions}[\text{auctionId}].\text{bidToken.balanceOf}(\text{prevBid.bidder})$$

  is decreased by $\text{bestBids}[\text{auctionId}].\text{amount} - \text{prevBid.amount}$.

The call $\text{auctionBid}(\text{auctionId}, \text{amount})$ reverts if

$$\text{amount} \leq \text{bestBids}[\text{auctionId}].\text{amount}.$$

Function Properties of `settleAuction`

Before calling `settleAuction`, let $\text{bid} = \text{bestBids}[\text{auctionId}]$.

A non-reverting call $\text{settleAuction}(\text{auctionId})$ will

- set $\text{bestBids}[\text{auctionId}].\text{bidder} = \text{address}(0)$

- transfer bid.amount to $\text{auctions}[\text{auctionId}].\text{seller}$ and transfer $\text{autions}[\text{auctionId}].\text{amount}$ to bid.bidder.