Dapper Dino's Documentation

1. Fossil Token contract

It's an ERC 20 token standard contract that is used to pay the rewards to the stakers who stake their NFT's in Dapper Dino.

Token Name : Fossil
Token Symbol : FOS
Token decimals : 18
Total Supply : 100000000000000000000000000

Contract functions :

1. name( ) : This function returns the name of the token.

2. symbol( ) : This function returns the symbol of the token.

3. totalSupply( ) : This function returns the total supply of the token.

4. balanceOf( ) : This function returns the balance (amount of token) of a particular address.

5. allowance( ) : This function returns the amount of tokens that a particular user can spend on behalf of the actual owner of the token.

6. transfer( ) : This function is used to send a specific amount of tokens from one address to some other address.

7. transferFrom( ) : This function is used to send a specific amount of tokens by one address on behalf of the actual owner of the tokens (Owner permits the address to spend on his behalf).

8. approve( ) : This function is used to give approval to some address by the owner so that specific address can spend the tokens on behalf of the actual owner.

9. changeContractOwner( ) : This function is used to change the contract owner and it can be called only by the contract owner.

10. burn( ) : This function is used to burn the tokens and it can be done by only those users who have BURNER ROLE permission.

11. mint( ) : This function is used to mint new tokens and it can be done by only those users who have MINTER ROLE permission.

2. Dino Pool contract

This contract is the pool where the NFT's of staker's are locked for a specific time and holds the rewards of the stakers and can be claimed by the stakers after the lock up period ends.

Important points :

1. The minimum time period for staking is 10 minutes.
2. The maximum time period for staking (maximum lock duration) is 1 year (365 days).
3. Rewards can be claimed after the admin has distributed rewards.
4. While staking NFT , the user has to give approval of NFT to DinoPool contract so that it can transfer NFT to its contract and lock it and release it while unstaking.
5. DinoPool contract gets the reward amount as per the amount staked by the user's and for the specific time period from the LiquidityMiningManager contract which is handled by the admin.
6. Deposit token is NFT & reward token is Fossil Token for this contract.
7. Reward distribution cycle is 1 day (means rewards are distributed after every 24 hours).
8. Weight of this pool is 100%.

Token Name :Staked Dino Token
Token Symbol : SDT
Token decimals : 18

Contract functions :

1. claimRewards( ) : This function is used to claim rewards (Fossil Token) after the staking of NFT.

2. deposit( ) : This function is used to stake the NFT of the user. It takes an array of tokenId, staking duration & receiver address as the parameters.

3. withdraw( ) : This function is used to withdraw the staked NFT's. Withdrawal can be performed after the staking period is over. It takes an array of depositId & receiver address as the parameters.

4. balanceOf( ) : This function returns the balance (amount of acknowledgement token) of a particular address.

5. getDepositsOf( ) : This function returns all the deposits (staked NFT) made by the particular address. This function takes the user's address as a parameter.

6. getDepositsOfLength( ) : This function returns the length of all the deposits (staked NFT) made by the particular address. This function takes the user's address as a parameter.

7. getMultiplier( ) : This function returns the maximum amount as a reward for staking for a particular time period. It takes lock duration as a parameter in milliseconds.

8. getTotalDeposit( ) : This function returns the total deposits (staked NFT) in quantity made by the particular address. This function takes the user's address as a parameter.

9. withdrawableRewardsOf( ) : This function returns the rewards that are claimable by a specific user. It gets calculated once the admin distributes rewards else it returns 0. This function takes the user's address as a parameter.

10. withdrawnRewards( ) : This function returns the rewards claimed till now by the user. This function takes the user's address as a parameter.

11. cumulativeRewardsOf( ) : This function returns the amount of funds that an address has earned in total till now as a reward. It takes the user's address as a parameter.

12. _distributeRewards( ) : This function is an internal function and is called by LiquidityMiningManager contract only. It distributes the rewards to all the user's as per their staked amount.

13. setMinimumLockDuration( ) : This function is used to change the minimum lock duration of staking. By default it is set to 10 minutes. Only the contract owner is allowed to change this value. This function takes lock duration in seconds as a parameter.

14. setMaximumLockDuration( ) : This function is used to change the maximum lock duration of staking. By default it is set to 365 days. Only the contract owner is allowed to change this value. This function takes lock duration in seconds as a parameter.

15. setMaximumBonus( ) : This function is used to change the maximum bonus of staking which is used by the getMultiplier function. By default this value is 1 Fossil. This function takes a max bonus value (in wei) as a parameter.

16. changeContractOwner( ) : This function is used to change the contract owner and it can be called only by the contract owner.

17.setMaximumNftStakingAllowed( ) : This function is used to set the maximum NFT staking quantity in the pool. By default it is 20 and it can be called only by the contract owner.

3. LiquidityMiningManager contract

This contract is responsible for distributing the rewards to all the stakers .It handles the pools added for staking and what's the weight of the pool. Roles to the users are provided by this contract only like Governor role, Distributor role etc. This contract has got approval from FossilToken contract so that it can transfer any amount of Fossil tokens as a reward whenever needed to distribute. It controls the rewards generated per second. It acts as an admin panel for the Dapper Dino.

Important points :

1. Rewards per second is set to 0.005 Fossil Token in this contract.
2. Reward token is Fossil Token.
3. It contains only one Staking pool, the Dino pool that holds NFT's.
4. The weight of the Dino pool is set to 100%.
5. Only contract deployers have been given the role of Governor & Reward Distributor.
6. This contract has 100% allowance of Fossil Tokens.
7. This contract can change rewards per second & and adjust the weight of the pools.
8. Reward source is the owner's wallet address (deployer holding all Fossil Tokens).

Contract functions :

1. addPool( ) : This function is used to add a new pool for staking. It can be called by the user who has got the role of Governor (admin). It takes the pool contract address & weight percent as a function parameter.

2. adjustWeight( ) : This function is used to change the weight of the pool. It takes pool id and weight as the parameters. It can be called by the user who has got the role of Governor (admin).

3. distributeRewards( ) : This function is used to distribute rewards to the stakers. This function transfers the rewards to the respective pool according to the pool shares & then user's can claim it. Users having the role as Distributor can only call this function.

4. grantRole( ) : This function is used to grant roles to the user's and it can be called by the admin only. It takes the role name and the user's address as the parameter.

5. removePool( ) : This function is used to remove the pool.It takes the pool id as a parameter. It can be called by the user having the Governor role.

6. setRewardPerSecond( ) : This function is used to set the rewards generated per second. It takes the reward amount as a parameter. It can be called by the user having the Governor role.

7. getPools( ) : This function returns all the pools.

8. revokeRole( ) : This function is used to take back the specific role means that particular address will now no longer have this role power. This function takes the role of bytes and user address as the parameter. This method can be called by the user having admin role permission.

4. UtilityManager contract

This contract is responsible for tracking all the details like total amount locked, no. of NFT staked, claimed rewards, unclaimed rewards, pending rewards. This contract acts as a Utility contract to store the values which are to be used in the UI for pool & token details purposes.

Important points :

1. UtilityManager values get updated accordingly as per the other contracts functions like claiming rewards, staking , unstaking etc.
2. All the functions of this contract can be called only by the existing contracts of this project. No other contract or user can call these functions.

Contract functions :

1. updateClaimedAmount( ) : This function is used to update the claimed amount as per the reward amount of each deposit. It takes the claimed amount as the parameter.

2. updatePendingRewards( ) : This function is used to update the pending rewards. It takes amount and operation (1/0) as a parameter. Here 1 means pending rewards have to be incremented and 0 means pending rewards have to be decrement.

3. updateStakedAmount( ) : This function is used to update the total staked amount. It takes the amount staked as the parameter.

4. updateTotalValueLockedAmount( ) : This function is used to update the amount locked in a pool that is staked. It takes amount and operation (1/0) as a parameter. Here 1 means amount has to be incremented and 0 means amount has to be decremented.

5. updateUnclaimedAmount( ) : This function is used to update the unclaimed rewards by the stakers. It takes amount and operation (1/0) as a parameter. Here 1 means amount has to be incremented and 0 means amount has to be decremented.

6. details( ) : This function returns the complete details like APR, staked amount, total value locked, claimed rewards and unclaimed rewards.

7. setContractAddress( ) : This function is used to initialise other contract addresses so that only existing contracts can call the methods of this contract .

5. Dino contract

This contract is a NFT contract that is ERC 721 standard. User's can purchase these NFT's by providing Ether as a collateral.

Token Name : Dapper Dinos
Token Symbol : DINO

Contract functions :

1. name( ) : This function returns the name of the token.

2. symbol( ) : This function returns the symbol of the token.

3. totalSupply( ) : This function returns the total supply of the token.

4. balanceOf( ) : This function returns the balance (amount of token) of a particular address.

5. walletOfOwner( ) : This function returns the tokens owned by the caller.

6. tokenURI( ) : This function is used to return the URI of specific token details.

7. pause( ) : This function is used to pause the sale.

8. approve( ) : This function is used to give approval to some address by the owner so that specific address can spend the tokens on behalf of the actual owner.

9. tokenOfOwner( ) : This function returns the total token count owned by the caller.

10. ownerOf( ) : This function is used to retrieve the owner of a particular token id.

11. getApproved( ) : This function is used to check whether a particular token is approved to any address or not. It returns true or false according to the approval.

12. mint( ) : This function is used to mint new NFT's and can only be called by the owner. Maximum 20 tokens can be minted at a time.

13. safeTransferFrom( ) : This function is used to send specific tokens by one address on behalf of the actual owner of the tokens (Owner permits the address to spend on his behalf).

14. setApprovalForAll( ) : This function is used to provide approval for all the tokens holded by a particular address to another address.

15. setBaseURI( ) : This function is used to set base URI for all tokens.

16. setPrice( ) : This function is used to set the price of a particular token.

17. transferOwnership( ) : This function is used to change the ownership of the token from one address to another.

18. withdrawAll( ) : This function is used to withdraw all tokens & can be called only by the owner. This method is payable.

6. RewardsPool contract

This contract is responsible to hold the rewards that are claimed by the user's. It stores it till the vesting period gets over and after that users can withdraw their rewards.

Important points :

1. Vesting period is set to 12 months which means after claiming rewards by the user, the rewards get locked in this pool for 12 months and after that only rewards can be withdrawn.

Contract functions :

1. setContractAddresses( ) : This function is used to set other contract addresses like DinoPool and Fossil Token. Only contract owners can call this function..

2. addReward( ) : This function is used to add the reward details in the contract like reward amount, start time of vesting period & end time of vesting period. This function can be called by LiquidityMiningManager contract only. This function takes the amount and address of the user as the parameter.

3. withdraw( ) : This function is used to withdraw the claimed rewards once vesting period is over. This function takes the address of the reward receiver and withdraw id as the parameter.

4. getUserRewardsList( ) : This function is used to get all the details of the rewards claimed till now. This function takes the address of the reward receiver as the parameter.

5. setVestingPeriod( ) : This function is used to change the vesting period. By default it is set to 1 year. This function takes the vesting period time in seconds as the parameter. It can be called only by the contract owner.

6. changeContractOwner( ) : This function is used to change the contract owner. This function takes the new owner's address as a parameter. It can be called only by the contract owner only.

Staking Procedure

1. Users need to select NFT (can select multiple NFT's at a time)
2. Then set the staking lock duration time period.
3. Then give approval to the DinoPool contract.
4. Then stake the NFT and NFT will be transferred to DinoPool and gets locked..

Unstaking Procedure

1. Users need to check that the locking period is over.
2. If yes, the user can unstake the NFT (multiple NFT's allowed) otherwise not.
3. On unstaking the NFT is transferred from DinoPool to the user's address.

Claiming Rewards Procedure

1. Once the admin has distributed the rewards then users can claim rewards otherwise not.
2. The reward amount (Fossil Token) is transferred to the RewardsPool contract.
3. After claiming the reward, the rewards get locked into RewardsPool for a vesting period of 12 months.
4. Once the vesting period is over, users can call the withdraw method of RewardsPool contract and receive their rewards in their wallet.