Uniswap-v2-core Code review

Uniswap is a protocol on Ethereum for exchanging tokens, just like a decentralised exchanged platform.

Table of content

Pragma solidity =0.5.16;

Imports

IuniswapV2Pair.sol: The pair contract facilitates all token swaps or trade

UniswapV2ERC20.sol

Libraries

SafeMath.sol

UQ112x112.sol

Interfaces

IERC20.sol: This is the interface interacting with the ERC20 token.

IuniswapV2Factory.sol: The factory contract is used to add new tokens to the **platform**

IUniswapV2Callee.sol

Contract

UniswapV2Pair: name of our parent contract

Inheriting

IUniswapV2Pair

UniswapV2ERC20

State Variables

MINIMUM-LIQUIDITY

SELECTOR

State variable

Factory

Token0

Token1

price0CumulativeLast

price1CumulativeLast

kLast

Private function (uint)

Reserve0: This signifies the liquidity reserve that stores trade fees charges by Uniswap on a % of trade. Whenever a liquidity provider decides they want to exit, they receive a portion of the total fees from the reserve relative to their staked amount in that pool. The token they received which keeps a record of what stake they’re owed is then destroyed.

reserve1

blockTimestampLast

Unlocked: The uniswap token unlocked after a specific period of time.

Events

Mint: enables senders mint tokens, specifying the amount
burn
swap
sync

Functions

getReserves:

The getReserves functions takes in no argument with the visibility function called public. This function also prevents state variables in the function to be called. It also returns 3 parameters _reserve0, _reserve, and _blockTimestampLast.

_safeTransfer:

This function takes in 3 arguments token(address), to(address) and value(uint). It is also only visible within the contract.

Initialize:

The initialize function accepts the _token0 and _token1 variable as an address datatype. This function can only be called by an external contract(in this case, by the factory contract), hence the use of the “external” visibility function. This function also uses a require statement that enables the person that calls the function to check the token factory. Called once by the factory at time of deployment.

Update :

Accepts 4 arguments balance0,balance1, _reserve0, _reserve1 as unsigned integers. This function can only be called by functions in uniSwapV2pair contract. It also requires that balance values is less than or equals uint112, else it should return an overflow.

mintFee:

This function takes on reserve0, and _reserve1 argument that returns the output of the mint fee. If fee is on, mint liquidity is equivalent to 1/6th of the growth in sqrt.

Mint:

This function is called from a contract which performs important safety checks

Burn:

This low-level function should be called from a contract which performs important safety checks.

Swap:

Ths low-level function should be called from a contract which performs important safety check to swap tokens.

Skim: forces balance to match reserves.

Sync: forces reserves to match balances.

Eniola Agboola.