한 번에 끝내는
블록체인 개발 A to Z

Chapter 3

Defi 기초 컨셉 구현

Chapter 3

Defi 기초 컨셉 구현

구현 계획 및 유니스왑 설명

이번 챕터는 이렇게 진행됩니다.

Clip 1
구현 계획 및 유니스왑

Clip 2
ERC20, Wrapped Token의 이해

Clip 3
SmartContract 개발 및 테스트 환경

Clip 4
AMM 개념 및 구현

Clip 5,6
CPMM 구현

Clip 7
Swap 기능 구현

Clip 8
LP 토큰, 수수료 구현

Clip 10
Factory SmartContract 구현

구현 목표

유니스왑v1 구현

유니스왑v1에는 CPMM, LP토큰, ETH<->ERC20 스왑, 유동성 공급, 수수료 보상 등
디파이의 스왑과 과련된 모든 개념이 들어있다.

구현 범위

SmartContract 구현 및 배포

유니스왑v1의 동작 방식과 함께 Solidity 소스코드를 분석하고 직접 구현한다.
구현한 SmartContract를 이더리움 테스트 네트워크에 배포해본다.

SmartContract 테스트 코드 작성

작성한 SmartContract의 테스트코드를 작성하여 배포하기 전에 구현한 함수의 동작을 빠르고 편리하게 테스트해본다.

프론트앤드 연동 구현

프론트앤드에서 메타마스크 지갑 연결과 배포한 SmartContract의 함수를 호출하여 동작하는 것을 확인한다.

유니스왑v1 프론트앤드 연동

실제 유니스왑v1의 프론트앤드 소스코드를 연동해본다.

SmartContract 구현,
테스트코드 작성 및 배포

VisualStudio Code
개발 IDE

Solidity
개발 언어

HardHat
배포 툴

Chai
테스트 툴

개발 환경

Ethereum
BlockChain Network

All That Node
Ethereum RPC Node

배포 네트워크

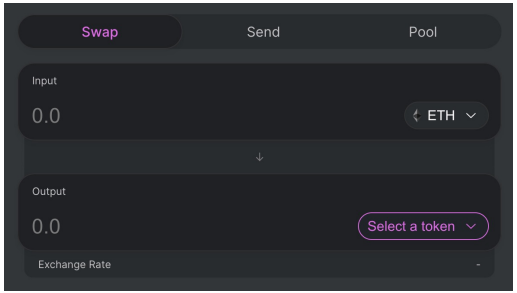
Frontend 연동 구현

VisualStudio Code
개발 IDE

React
프론트앤드 프레임워크

MetaMask
블록체인 지갑

개발 환경



소스코드: https://github.com/Uniswap/v1-interface

유니스왑v1

유니스왑v1

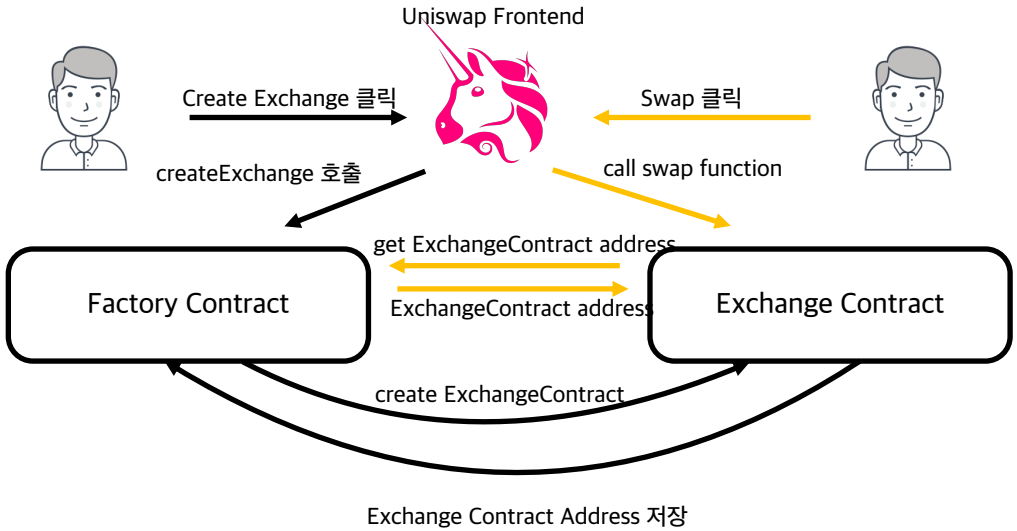
- 최초의 CPMM 프로젝트이다.

$$x * y = k$$

- 이더리움을 기축통화로 사용한다. ERC20/ERC20 페어는 지원하지 않는다.

  ERC20 <-> ERC20을 위해서는 ERC20을 이더리움으로 바꾸고 다시 이더리움을 ERC20으로 바꾼다.
- 유동성 공급자들에게 매번 거래 금액의 0.3% 수수료를 보상으로 지급한다.
- 유니스왑이 가져가는 수수료는 없다.
- 거버넌스 토큰(UNI토큰) 및 스테이킹 기능은 없다.
- Vyper라는 SmartContract 개발 언어로 개발되었다.(Not Solidity)
- Exchange SmartContract, Factory SmartContract 두 개의 SmartContract로 동작한다.
  - Exchange SmartContract: 유동성 공급, 제거, LP토큰, AMM 함수 등이 존재
  - Factory SmartContract: 토큰페어(Exchange Contract) 생성, 조회 등의 함수 존재

유니스왑v1 SmartContract 구성



GOAL

- 유니스왑v1의 Exchange, Factory SmartContract를 구현하고 이해한다.
- $x * y = k$ 공식이 사용되는 CPMM을 구현하고 동작 방식 및 과정을 이해한다.
- CPMM알고리즘으로 ETH<->ERC20, ERC20<->ETH<->ERC20 스왑을 구현한다.
- 유동성 공급 및 제거, LP토큰 발행을 구현하고 동작 및 과정을 이해한다.
- 유동성 공급자들에게 매번 거래 금액의 0.3% 수수료가 보상으로 지급되는 로직을 이해한다.
- Impermanent Lose, Slippage를 수식과 소스코드로 이해한다.