

한 번에 끝내는 블록체인 개발 A to Z

Chapter 3

Defi 기초 컨셉 구현

Chapter 3

Defi 기초 컨셉 구현

유니스왑v1 CPMM 구현(4)

Goal

LP에게 제공하는 보상인 수수료를 구현한다.

수수료 구현에 대한 아이디어

- 유동성 공급자들(LP)에게 수수료를 지급하는 방식은 다양 할 수 있다.

: 수수료는 뭐로 지급할까? ETH로 줄까? ERC20으로 줄까? LP토큰으로 줄까?

- 트레이더에게 수수료를 부과하는 방식은 다양 할 수 있다.

: 수수료를 포함하여 InputAmount를 입력하도록 할까? 수수료를 제하고 OutputAmount를 줄까?

수수료 구현에 대한 아이디어

- 유니스왑에서는 LP들에게 ETH, ERC20 모두 보상으로 제공할 수 있는 로직을 가지고 있다.
 - 트레이더가 지급한 수수료 만큼 유동성 풀의 토큰 개수가 증가한다.
- 따라서 LP들이 유동성을 제거 할 때 공급 한 개수 보다 많은 토큰을 가져 갈 수 있다.
 - 트레이더는 수수료를 제한 개수만큼 OutputAmount를 받게 된다.
 - 유니스왑은 트레이딩 마다 0.3%의 수수료를 부가한다.

수수료 구현에 대한 아이디어

```
function getOutputAmount(uint256 inputAmount, uint256 inputReserve, uint256 outputReserve)
    uint256 numerator = (inputAmount * outputReserve);
    uint256 denominator = (inputReserve + inputAmount);
    return numerator / denominator;
}
```



```
function getOutputAmount(uint256 inputAmount, uint256 inputReserve, uint256 outputReserve) public
    uint256 inputAmountWithFee = inputAmount * 99;
    uint256 numerator = (inputAmountWithFee * outputReserve);
    uint256 denominator = (inputReserve * 100 + inputAmountWithFee);
    return numerator / denominator;
}
```

수수료 구현에 대한 아이디어

```
function getOutputAmount(uint256 inputAmount, uint256 inputReserve, uint256 outputReserve) public  
    uint256 inputAmountWithFee = inputAmount * 99;  
    uint256 numerator = (inputAmountWithFee * outputReserve);  
    uint256 denominator = (inputReserve * 100 + inputAmountWithFee);  
    return numerator / denominator;  
}
```

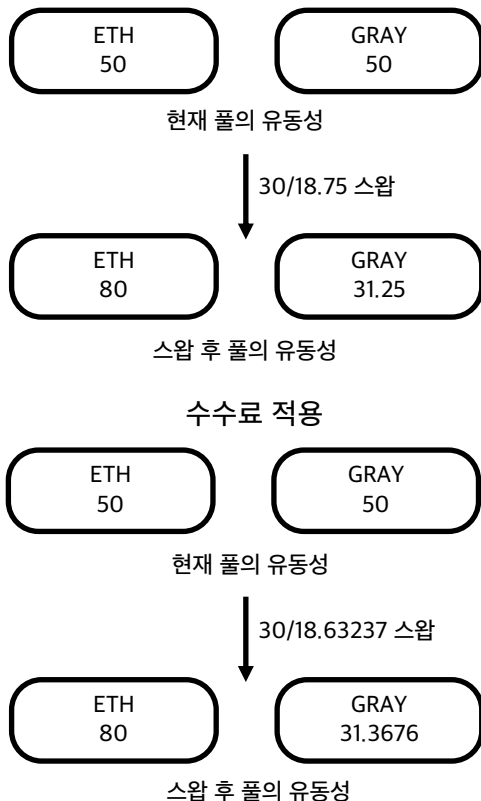
- 트레이더에게 수수료를 제외하고 토큰을 스왑해준다.
- 예를 들어 100개의 토큰을 InputAmount로 넣었는데 99개의 InputAmount로 OutputAmount를 계산한다.
그만큼 OutputAmount가 줄어들어 내가 받게 되는 토큰의 개수가 줄어든다.
- OutputAmount 수수료 만큼 Output에 해당하는 토큰의 Reserve가 증가하는 효과가 있다.

수수료 구현에 대한 아이디어

- 수수료가 0인 상황에서는 ETH:50, GRAY:50개인 풀에서 ETH30개를 스왑하면 GRAY 18.75개를 받게 된다.

$$\Delta y = \frac{y\Delta x}{x+\Delta x} \text{에 의해 } \Delta y = \frac{50*30}{50+30} \text{ 따라서 } \Delta y \text{는 } 18.75$$

- 수수료가 1%인 상황에서는 $\Delta y = \frac{50*30*(100-1)}{(50*100)+30*(100-1)}$ 따라서 Δy 는 18.6323713927227
- $18.75 - 18.6323713927227 = 0.1176286072773$ 으로 약 0.1176개를 수수료로 지불한 것이 된다.
- 수수료가 없을 때는 스왑 후 GRAY풀에는 31.25개가 있지만 수수료 1%를 적용하면 31.3676개로 0.1176개가 추가 된다.
- 이는 LP들이 유동성을 제거했을 때 가져 가게 되는 수수료이다.



구현 및 테스트 (수수료 적용)

- git clone https://github.com/GrayWorld-io/lec_fc_defi
- cd lec_fc_defi
- git reset --hard 768f9d19287f435cf354286558656866c537d55f

정리

- 기본적인 CPMM 설명 및 구현이 완료되었다. 챕터3의 핵심 강의가 끝났다.
- 우리는 유동성 공급을 하고 LP 토큰을 받는걸 구현 했다.
- CPMM 공식을 활용해 스왑을 구현하였다.
- 스왑에 의해 유동성 풀의 토큰 비율이 변하여 비영구적 손실이 발생하는 것을 확인 했다.
- 스왑을 할 때 트레이더들에게 수수료를 부가하여 LP에게 보상을 주는 로직을 적용했다.
- 유동성을 제거를 구현하였고 제거 후 LP에게 보상이 들어온 것을 확인하였다.

다음 강의

- 토큰 페어에 대한 정보를 저장하는 Factory SmartContract를 구현해야 한다.
- Factory Contract와 Exchange Contract가 연동되는 것을 이해하기 위하여 '인터페이스'에 대해서 이해한다.