

한 번에 끝내는 블록체인 개발 A to Z

Chapter 3

Defi 기초 컨셉 구현

Chapter 3

Defi 기초 컨셉 구현

유니스왑v1 CPMM 구현(3)

Goal

- 유동성 공급 함수를 개선한다.
- LP토큰을 적용해본다.

유동성 공급

- 유동성을 공급하는 상황에는 두 가지의 상황이 있다.
 - 1) 유동성이 0인 상황. 이는 풀에 토큰 개수가 하나도 없는 경우이며, 풀이 처음 만들어진 경우 뿐이다.
 - 2) 유동성이 있는 상황. 추가로 유동성을 공급하는 경우이다.
- 유동성을 공급하면 LP토큰을 받는다.
- 기존에 구현 했던 addLiquidity 함수를 개선해보자.

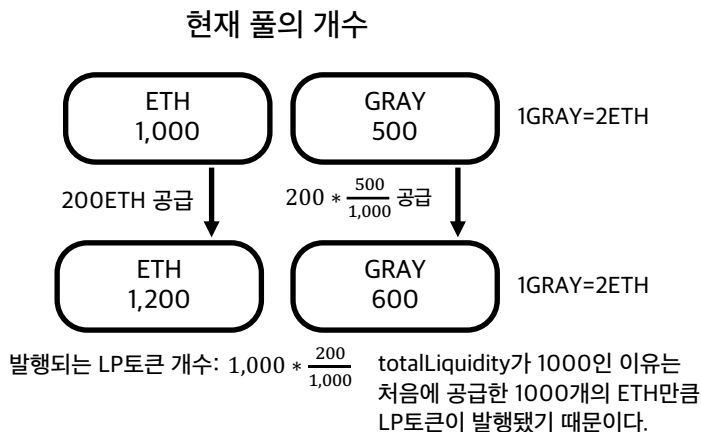
```
contract Exchange {  
  
    function addLiquidity(uint256 _tokenAmount) public payable {  
        | token.transferFrom(msg.sender, address(this), _tokenAmount);  
    }  
  
}
```

구현 및 테스트(CPMM)

- git clone https://github.com/GrayWorld-io/lec_fc_defi
- cd lec_fc_defi
- git reset --hard 2d338430b3d2234c580b4d2ee4450f16d4b6580d

유동성 공급 및 LP토큰 발행

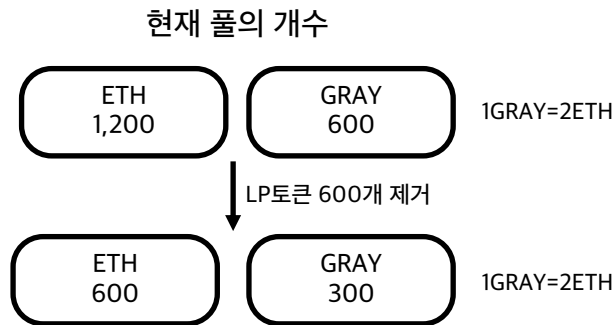
```
function addLiquidity(uint256 _maxTokens) public payable returns (uint256) {
    uint256 totalLiquidity = totalSupply();
    if (totalLiquidity > 0) {
        uint256 ethReserve = address(this).balance - msg.value;
        uint256 tokenReserve = token.balanceOf(address(this));
        uint256 tokenAmount = msg.value * tokenReserve / ethReserve;
        require(_maxTokens >= tokenAmount);
        require(token.transferFrom(msg.sender, address(this), tokenAmount));
        uint256 liquidityMinted = totalLiquidity * msg.value / ethReserve;
        _mint(msg.sender, liquidityMinted);
        return liquidityMinted;
    } else {
```



- ETH와 페어로 공급되는 토큰의 양은 1:1 비율로 공급되어야 한다. 공급 후에도 풀에 비율은 같아야 하기 때문이다. 따라서 공급되는 ETH 개수에 풀에 존재하는 토큰과 이더의 비율을 곱하면 공급하는 토큰의 양을 구할 수 있다.
- 유니스왑v1은 발행되는 LP토큰의 개수를 ETH의 개수를 기준으로 한다. 내가 공급하는 ETH의 개수와 전체 풀에 존재하는 ETH 비율을 곱하여 발행하는 LP토큰 개수를 구할 수 있다.

유동성 공급 및 LP토큰 발행

```
function removeLiquidity(uint256 _lpToken) public returns (uint256, uint256) {  
    uint256 totalLiquidity = totalSupply();  
    uint256 ethAmount = _lpToken * address(this).balance / totalLiquidity;  
    uint256 tokenReserve = token.balanceOf(address(this));  
    uint256 tokenAmount = _lpToken * tokenReserve / totalLiquidity;  
  
    _burn(msg.sender, _lpToken);  
  
    payable(msg.sender).transfer(ethAmount);  
    token.transfer(msg.sender, tokenAmount);  
    return (ethAmount, tokenAmount);  
}
```



- 유동성 공급자가 유동성을 제거 할 때에는 LP토큰을 소각하고, ETH와 토큰을 돌려 받는다.
- 돌려받는 ETH와 토큰의 개수는 내가 회수하고자 하는 LP토큰의 개수와 전체 풀의 비율 만큼 돌려 받는다.

질문

CPMM 공식에서 $xy=k$ 로 k 는 변하지 않는다고 했는데 유동성 공급으로 인해 x,y 값이 증가하면 k 값이 증가하지 않나?



증가하는 것이 맞다.

CPMM은 스왑과 같이 풀의 토큰 비율이 변하는 경우에도 곱의 값이 일정한 것이다.

유동성 공급으로 인한 x,y 변화는 비율이 일정하기 때문에 k 값은 증가하게 된다.

다음 강의

유동성 공급, 제거와 스왑을 통해 Impermanent Lose(비영구적 손실)을 확인해본다.