Audit

git :https://github.com/ooMia/Upside_DEX_solidity/blob/main/src/Dex.sol

commit: `0f49cc9c5179fd611c90028dc67f8a10378d23b8`

ID	요약	위험도
MIA-001	`addLiquidity` 함수에서 최소 유동성 요구사항 부재와 단순한 LP 토큰 발행 로직으로 인해 불균형한 토큰 분배가 가능	High

Description

`addLiquidity` 함수를 통해 초기 유동성을 소액으로 제공한 후, 대규모 유동성이 추가될 때 문제가 발생합니다. 총 유동성 풀의 크기가 증가하면, 초기 유동성 제공자가 자신의 LP 토큰을 제거할 때 토큰 비율이 여전히 높아 제공한 금액에 비해 훨씬 많은 양의 LP 토큰을 받게 됩니다.

Impact

High

초기 유동성 제공자가 불균형적으로 큰 이익을 얻을 수 있어, 다른 사용자들에게 경제적 손실을 초래할 수 있습니다.

Recommendation

- 초기 유동성 제공 시 최소 금액을 설정하여 공격을 방지
- 유동성 추가/제거 시 가격 영향을 확인하는 로직을 구현

ID	요약	위험도
MIA-002	`removeLiquidity` 함수에서 계산 과정 중 오차 발생으로 인하여 LP토큰을 탈취 당할 수 있음	Medium

Description

`removeLiquidity` 함수에서 오차가 발생합니다. 공격자는 이를 악용하여 유동성을 제공하고 제거하는 과정에서 매우 작은 양의 유동성을 제거할 때 발생하는 오차를 이용합니다. 그 결과, 공격자는 원래 제공한 유동성보다 더 많은 양의 토큰을 획득할 수 있습니다.

Impact

Medium

공격자가 반복적인 소액 거래를 통해 시스템에서 추가적인 토큰을 얻을 수 있지만, 실행에 많은 시간과 가스 비용이 필요합니다.

Recommendation

- 최소 유동성 제거 금액을 설정하여 매우 작은 금액의 거래를 제한
- **OpenZeppelin**의 `SafeMath`와 같은 검증된 수학 라이브러리를 사용하여 정밀한 계산을 수행

ID	요약	위험도
MIA-003	`swapX, swapY` 풀의 실제 상태가 업데이트 되고 있지 않음	Critical

Description

`swapX`와 `swapY` 함수에서 현재 유동성 풀의 상태를 업데이트하는 로직이 누락되어 있습니다. 이로 인해 `swap`을 여러 번 실행해도 항상 동일한 양의 토큰이 전송됩니다.

Impact

Critical

유동성 풀의 상태를 업데이트하고 있지 않아 사용자들이 이익 또는 손해를 보게됩니다.

Recommendation

- swap 함수 내에서 `balanceX`와 `balanceY`를 정확히 업데이트하도록 로직을 수정