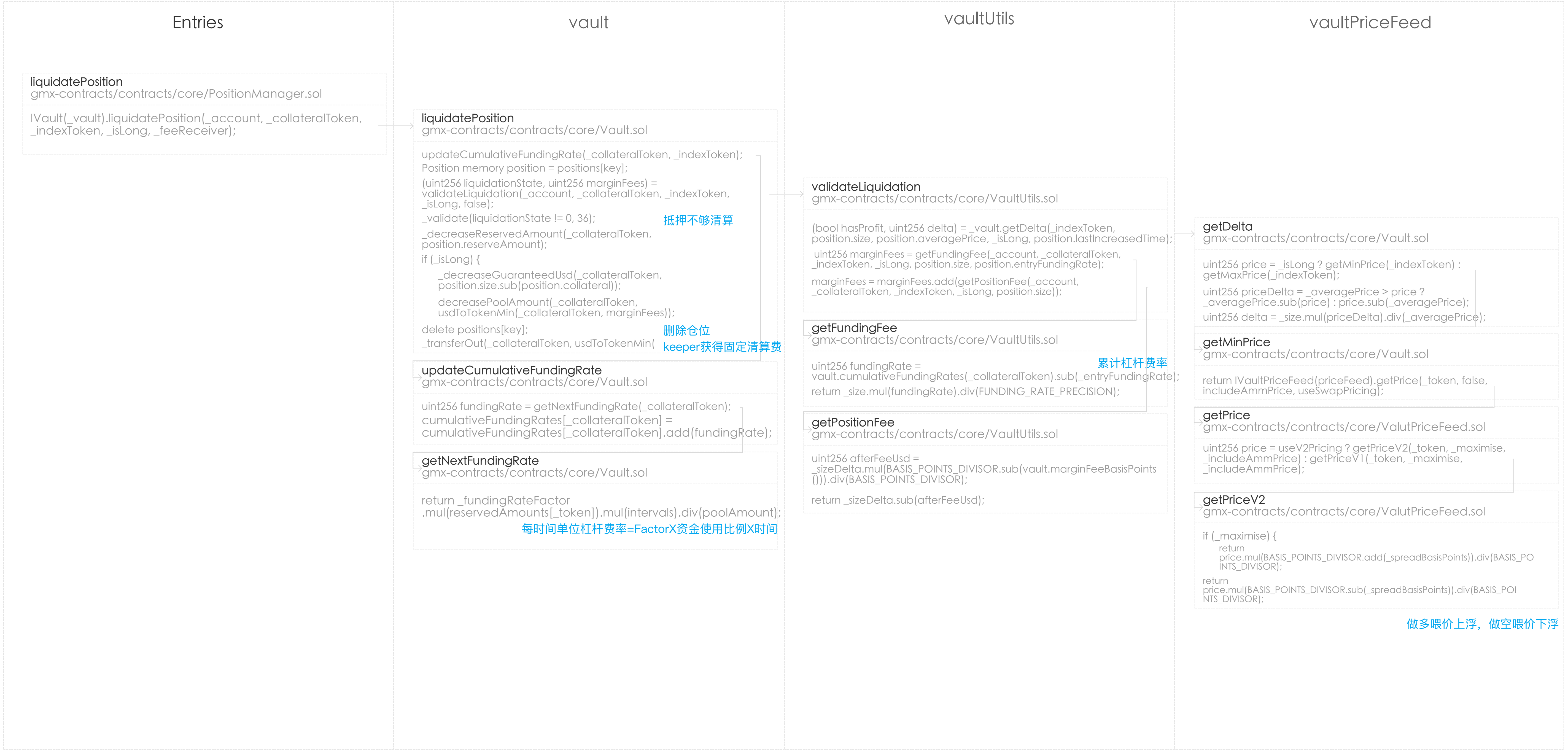
gmx No.2 liquidate_position

https://github.com/xiaoyu1998

Resolution:2560×1440(2K)
version: 6d393fa8ee26b374856f6b603cb96241ca77b3cb



Entries

liquidatePosition
gmx-contracts/contracts/core/PositionManager.sol

```
IVault(_vault).liquidatePosition(_account, _collateralToken,
_indexToken, _isLong, _feeReceiver);
```

vault

liquidatePosition
gmx-contracts/contracts/core/Vault.sol

```
updateCumulativeFundingRate(_collateralToken, _indexToken);
Position memory position = positions[key];
(uint256 liquidationState, uint256 marginFees) =
validateLiquidation(_account, _collateralToken, _indexToken,
_isLong, false);
_validate(liquidationState != 0, 36);
_decreaseReservedAmount(_collateralToken,
position.reserveAmount);
if (_isLong) {
    _decreaseGuaranteedUsd(_collateralToken,
    position.size.sub(position.collateral));
    decreasePoolAmount(_collateralToken,
    usdToTokenMin(_collateralToken, marginFees));
delete positions[key];
_transferOut(_collateralToken, usdToTokenMin(
```

抵押不够清算

删除仓位

keeper获得固定清算费

updateCumulativeFundingRate
gmx-contracts/contracts/core/Vault.sol

```
uint256 fundingRate = getNextFundingRate(_collateralToken);
cumulativeFundingRates[_collateralToken] =
cumulativeFundingRates[_collateralToken].add(fundingRate);
```

getNextFundingRate
gmx-contracts/contracts/core/Vault.sol

```
return _fundingRateFactor
.mul(reservedAmounts[_token]).mul(intervals).div(poolAmount);
```

每时间单位杠杆费率=FactorX资金使用比例X时间

vaultUtils

validateLiquidation
gmx-contracts/contracts/core/VaultUtils.sol

```
(bool hasProfit, uint256 delta) = _vault.getDelta(_indexToken,
position.size, position.averagePrice, _isLong, position.lastIncreasedTime);
 uint256 marginFees = getFundingFee(_account, _collateralToken,
_indexToken, _isLong, position.size, position.entryFundingRate);
marginFees = marginFees.add(getPositionFee(_account,
_collateralToken, _indexToken, _isLong, position.size));
```

getFundingFee
gmx-contracts/contracts/core/VaultUtils.sol

```
uint256 fundingRate =
vault.cumulativeFundingRates(_collateralToken).sub(_entryFundingRate);
return _size.mul(fundingRate).div(FUNDING_RATE_PRECISION);
```

累计杠杆费率

getPositionFee
gmx-contracts/contracts/core/VaultUtils.sol

```
uint256 afterFeeUsd =
_sizeDelta.mul(BASIS_POINTS_DIVISOR.sub(vault.marginFeeBasisPoints
())).div(BASIS_POINTS_DIVISOR);
return _sizeDelta.sub(afterFeeUsd);
```

vaultPriceFeed

getDelta
gmx-contracts/contracts/core/Vault.sol

```
uint256 price = _isLong ? getMinPrice(_indexToken) :
getMaxPrice(_indexToken);
uint256 priceDelta = _averagePrice > price ?
_averagePrice.sub(price) : price.sub(_averagePrice);
uint256 delta = _size.mul(priceDelta).div(_averagePrice);
```

getMinPrice
gmx-contracts/contracts/core/Vault.sol

```
return IVaultPriceFeed(priceFeed).getPrice(_token, false,
includeAmmPrice, useSwapPricing);
```

getPrice
gmx-contracts/contracts/core/ValutPriceFeed.sol

```
uint256 price = useV2Pricing ? getPriceV2(_token, _maximise,
_includeAmmPrice) : getPriceV1(_token, _maximise,
_includeAmmPrice);
```

getPriceV2
gmx-contracts/contracts/core/ValutPriceFeed.sol

```
if (_maximise) {
    return
    price.mul(BASIS_POINTS_DIVISOR.add(_spreadBasisPoints)).div(BASIS_PO
    INTS_DIVISOR);
return
price.mul(BASIS_POINTS_DIVISOR.sub(_spreadBasisPoints)).div(BASIS_POI
NTS_DIVISOR);
```

做多喂价上浮，做空喂价下浮