DEFI Crash Course: Money Markets Exercise 1 - Aave V3

Intro

Your goal is to create and test a contract that interacts with the AAVE V3 protocol, deposits USDC, and borrows DAI.

Implement 4 functions:

1. `depositUSDC` - Deposit USDC as collateral to AAVE
2. `withdrawUSDC` - Withdraw the deposited USDC from AAVE
3. `borrowDAI` - Borrow DAI against the supplied USDC collateral (Variable Interest Mode)
4. `repayDAI` - Repay borrowed DAI

Note: This exercise is executed on an Ethereum mainnet Fork block number `16776127`. Everything is already configured in the `hardhat.config.js` file

Ethereum MAINNET Addresses

```
AAVE V3 Pool: 0x87870Bca3F3fD6335C3F4ce8392D69350B4fA4E2
USDC Token: 0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48
DAI Token: 0x6B175474E89094C44Da98b954EedeAC495271d0F
aUSDC Token: 0x98C23E9d8f34FEFb1B7BD6a91B7FF122F4e16F5c
Variable Debt DAI Token: 0xcF8d0c70c850859266f5C338b38F9D663181C314

Impersonated Account (Whale / Binance Hot Wallet):
0xf977814e90da44bfa03b6295a0616a897441acec
```

Accounts

- 0 - User

Tasks

Task 1 - Smart Contract Development

Complete all the open TODOS in the `./contracts/money-markets-1/AaveUser.sol` file

1. Define all the state variables, including:
   1. Aave Pool Contract
   2. DAI Contract and USDC Contract
   3. depositedAmount and borrowedAmount (to keep track of the deposited and borrowed amounts)
2. In the constructor: initialize the Aave Pool, DAI and USDC contracts.
3. Implement the depositUSDC function that allows you to supply USDC to AAVE.

4. Implement the withdrawUSDC function that allows you to withdraw USDC from AAVE.
5. Implement the borrowDAI function that allows you to borrow DAI form AAVE in a Variable Interest Mode.
6. Implement the repayDAI function that allows you to repay DAI to AAVE.

Task 2 – Tests

Complete all the open TODOS in the `./test/money-markets-1/tests.js` **file**

Useful Links

AAVE V3 Contracts

AAVE V3 Pool Interface

AAVE V3 Deployed MINNET Addresses