

Assure DeFi®

THE VERIFICATION **GOLD STANDARD**



Security Assessment

Gradient

Date: 20/09/2025

Audit Status: PASS

Audit Edition: Advanced+



Risk Analysis

Vulnerability summary

Classification	Description
● High	High-level vulnerabilities can result in the loss of assets or manipulation of data.
● Medium	Medium-level vulnerabilities can be challenging to exploit, but they still have a considerable impact on smart contract execution, such as allowing public access to critical functions.
● Low	Low-level vulnerabilities are primarily associated with outdated or unused code snippets that generally do not significantly impact execution, sometimes they can be ignored.
● Informational	Informational vulnerabilities, code style violations, and informational statements do not affect smart contract execution and can typically be disregarded.

Executive Summary

According to the Assure assessment, the Customer's smart contract is **Secured**.

Insecure

Poorly Secured

Secured

Well Secured

Scope

Target Code And Revision

For this audit, we performed research, investigation, and review of the Gradient contracts followed by issue reporting, along with mitigation and remediation instructions outlined in this report.

Target Code And Revision

Project	Assure
Language	Solidity
Codebase	<p>GradientMarketMakerFactory.sol [SHA256] - b7e757433553744c4d500fa6c66d37045626f96fa64388da0958da5b22e404ce</p> <p>GradientMarketMakerPoolV2.sol [SHA256] - e4ed0aa99b688b58971a319ed1eef911330c5c8aa9d789ad6db4d2c2ab42ba4b</p> <p>Fixed version [Commit]: 1ec62c4f2a759c5d485f9ade94e617c6175b69c8</p>
Audit Methodology	Static, Manual

Attacks made to the contract

In order to check for the security of the contract, we tested several attacks in order to make sure that the contract is secure and follows best practices.

Category	Item
Code review & Functional Review	<ul style="list-style-type: none">• Compiler warnings.• Race conditions and Reentrancy.• Cross-function race conditions.• Possible delays in data delivery.• Oracle calls.• Front running.• Timestamp dependence.• Integer Overflow and Underflow.• DoS with Revert.• DoS with block gas limit.• Methods execution permissions.• Economy model.• Private user data leaks.• Malicious Event log.• Scoping and Declarations.• Uninitialized storage pointers.• Arithmetic accuracy.• Design Logic.• Cross-function race conditions.• Safe Zeppelin module.• Fallback function security.• Overpowered functions / Owner privileges

AUDIT OVERVIEW



HIGH

1. Rewards sent when no providers exist become unclaimable (ETH & Token) [Partially Fixed ✓]

Contract: GradientMarketMakerPoolV2

Function: distributePoolFee, distributeTokenFee, _updatePoolRewards, _updateTokenRewards

Issue: If totalETH == 0 (or totalTokens == 0) at distribution time, the contract accepts the fee but does not update accRewardPerShare/accTokenRewardPerShare. There is no later catch-up, so those fees are never allocated effectively stranded in the contract.

Recommendation: Either forbid distributions when totalETH==0/totalTokens==0, or accrue undistributed fees in a pending bucket and add them to acc* the next time liquidity > 0, or immediately refund on zero-liquidity.

Fix: _updatePoolRewards: if totalETH == 0, it now refunds ethAmount to msg.sender and emits FeeRefunded.

_updateTokenRewards: if totalTokens == 0, it refunds tokenAmount (actually the actualReceived from distributeTokenFee) and emits FeeRefunded.

When totalETH == 0, _updatePoolRewards calls payable(msg.sender).call{value: ethAmount}() from distributePoolFee (which is not nonReentrant). It's gated by onlyRewardDistributor, but a compromised/malicious reward distributor could reenter.

Additional Fix: Mark distributePoolFee as nonReentrant and/or move the refund to happen after updating any relevant accounting, or convert to a pull-refund pattern.



MEDIUM

1. Hard dependency on EventAggregator can DoS pool creation [Fixed ✓]

Contract: GradientMarketMakerFactory

Function: constructor, setEventAggregator, createPool, createPoolWithLiquidity

Issue: Factory allows a zero or non-contract eventAggregator at deploy time. createPool* unconditionally calls eventAggregator.emitPoolCreated(). If the aggregator is zero, non-contract, or reverts, pool creation reverts at the end, halting deployments (availability risk).

Recommendation: Validate non-zero and code.length > 0 in constructor and setter. Wrap aggregator call in try/catch (best-effort) or gate it behind an optional flag. Rely on the on-chain PoolCreated event as the canonical signal and avoid making the aggregator a hard requirement.

Fix: Setter now requires non-zero and contract code:

```
if (address(_eventAggregator) == address(0) || address(_eventAggregator).code.length == 0) revert InvalidEventAggregator();
```

Both createPool and createPoolWithLiquidity wrap the external call in try/catch, so a bad aggregator can't revert pool creation:

```
try eventAggregator.emitPoolCreated(token, pool) { } catch { }
```

Constructor allows zero (optional aggregator) but validates code if non-zero. This removes the hard dependency/DoS vector.

2. Unsafe approve usage breaks non-standard ERC-20s during liquidity init [Fixed ✓]

Contract: GradientMarketMakerFactory

Function: createPoolWithLiquidity

Issue: Uses raw IERC20(token).approve(pool, amount) without SafeERC20 and without resetting to zero. Non-standard tokens that require zero-first or return false will cause failures/reverts, preventing initial liquidity.

Recommendation: Use SafeERC20.safeApprove(pool, 0) then safeApprove(pool, amount) or safeIncreaseAllowance. Alternatively, use a push transfer model or EIP-2612 permit to avoid approvals entirely.

Fix: Switched to SafeERC20.forceApprove, which handles tokens that require zeroing allowances first

3. Claim-Only Paths Blocked When pendingRewards == 0 [Fixed ✓]

Contract: GradientMarketMakerPoolV2

Function: claimETHRewardsWithProof, claimTokenRewardsWithProof

Issue: Early guard reverts if pendingRewards == 0, even when users have claimable rewards from position * acc - debt. This DoS's legitimate claims.

Recommendation: Remove the early pendingRewards checks. Compute rewards from acc - debt plus pendingRewards and revert only if the computed claimable amount is zero.

Fix: claimETHRewardsWithProof and claimTokenRewardsWithProof no longer gate on pendingRewards. They compute claimable and revert only if computed amount is zero.

4. Unbounded tokenDecimals Can Overflow Exponentiation [Fixed ✓]

Contract: GradientMarketMakerPoolV2

Function: constructor, _normalizeTo18Decimals, _denormalizeFrom18Decimals

Issue: Using $10^{** tokenDecimals}$ (and ± 18 shifts) can overflow/revert for malicious/non-standard decimals (like 250), causing deployment or runtime DoS

Recommendation: Enforce a sane upper bound (for ex require(tokenDecimals <= 24, "Unsupported token decimals");) before exponentiation, or use a safe, bounded power routine.

Fix: Added MAX_TOKEN_DECIMALS = 24 and UnsupportedTokenDecimals() in constructor. minTokenLiquidity and normalize/denormalize are now safe.

5. Fee-On-Transfer / Rebasing Tokens Break Accounting [Fixed ✓]

Contract: GradientMarketMakerPoolV2

Function: _addTokenLiquidity, _executeSellOrder, distributeTokenFee

Issue: Code credits tokenAmount without measuring actual received amount. Fee/rebase tokens deliver less, causing totalTokens to drift from real balance and later withdrawals to revert.

Recommendation: Use balance-delta accounting: record before = balanceOf(this), perform transfer, set received = balanceOf(this) - before, and use received for positions, totals, and reward updates or explicitly restrict to standard ERC-20s.

Fix: _addTokenLiquidity and _executeSellOrder now use balance-delta (balanceBefore/actualReceived). distributeTokenFee uses balance-delta and passes actualReceived to _updateTokenRewards.



LOW

1. Residual factory - pool allowance allows unintended future pulls (conditional) [Fixed ✓]

Contract: GradientMarketMakerFactory

Function: createPoolWithLiquidity

Issue: After adding token liquidity, the factory may retain a non-zero allowance for the pool. If tokens later appear at the factory (airdrops/mistakes) and the pool can transferFrom, it could pull them unexpectedly (depends on pool logic).

Recommendation: Reset allowance to zero immediately after use (safeApprove(pool, 0)), or prefer push-based transfers / one-shot permit patterns.

Fix: After adding token liquidity, the allowance is reset to zero

2. Pools can be created for non-contract token addresses (EOA) [Fixed ✓]

Contract: GradientMarketMakerFactory

Function: createPool, createPoolWithLiquidity

Issue: No check that token is a contract. A pool can be created for an EOA, resulting in an unusable/garbage pool and operational clutter.

Recommendation: Require token.code.length > 0 and keep the existing blockedTokens check.

Fix: Added code size check in both creation paths

3. Pool admin tied to factory owner at deploy time may drift from governance [Fixed ✓]

Contract: GradientMarketMakerFactory

Function: _getPoolBytecode

Issue: The pool constructor receives owner() snapshotted at deployment. If factory ownership changes later, newer pools will have a different admin than older ones, leading to governance inconsistency.

Recommendation: Pass the factory address and have pools reference factory.owner() dynamically, or add an explicit admin sync/transfer mechanism in pools and document governance behavior.

Fix: The pool constructor now receives address(this) (factory address) instead of owner()

4. Token Reward Counters Misclassified as “Removed” [Partially Fixed ✓]

Contract: GradientMarketMakerPoolV2

Function: _updateTokenRewards

Issue: Increases totalTokensRemoved when distributing token fees, even though tokens are added to the pool, corrupting telemetry/off-chain analytics.

Recommendation: Stop incrementing totalTokensRemoved here; introduce a dedicated tokenRewardsAdded (or similar) counter if needed.

Fix: _updateTokenRewards now increments totalTokenRewardsDistributed instead of totalTokensRemoved.

Remaining: In claimRewardsWithProof (combined claim), token rewards are paid but no metric counter is adjusted, in claimTokenRewardsWithProof the code increments totalTokensRemoved. Metrics still inconsistent across claim paths.

5. Event Aggregator Reverts Can Brick Core Flows [Fixed ✓]

Contract: GradientMarketMakerPoolV2

Function: Multiple getEventAggregator().emit* call sites

Issue: External aggregator calls are unconditional so if the aggregator misbehaves/upgrades and reverts, deposits, withdrawals, trades, or claims all revert.

Recommendation: Wrap aggregator emissions in try/catch or make them best-effort after state changes, optionally emit local events as fallback.

Fix: All aggregator emissions wrapped in try/catch.



INFORMATIONAL

1. Factory event duplication [Fixed ✓]

Contract: GradientMarketMakerFactory

Function: createPool, createPoolWithLiquidity

Issue: Emits PoolCreated and also eventAggregator.emitPoolCreated (twice in createPoolWithLiquidity, before/after liquidity). Not dangerous but can confuse indexers.

Recommendation: Emit once per pool creation (plus optional initial liquidity events) to avoid double counting.

Fix: The updated code no longer emits that event (only the aggregator call remains).

Testing coverage

During the testing phase, custom use cases were written to cover all the logic of contracts. *Check "Annexes" to see the testing code.

```
tests/test_maker_factory.py::test_set_event_agg RUNNING
Transaction sent: 0x154a8dc6341ae7423a9c43f09a3a47b42f67a30f94a87bca22cb3eccf4e8ce8a
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 0
    GradientRegistry.constructor confirmed Block: 1 Gas used: 1992969 (16.61%)
    GradientRegistry deployed at: 0x3194cBDC3dbcd3E11a07892e7bA5c3394048Cc87

Transaction sent: 0x05964b6a360dcc52c0b0366891a8d2ba3f9b3d46c01a327ba03fd3ce21e9bda
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 1
    MockEventAggregator.constructor confirmed Block: 2 Gas used: 451473 (3.76%)
    MockEventAggregator deployed at: 0x602C71e4DAC47a042Ee7f46E0aee17F94A3bA0B6

Transaction sent: 0x05b5054508b1a50eae6eb7e0e398c7da8698190bcf5c219ad788165c5c48f23a
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 2
    GradientMarketMakerFactory.constructor confirmed (InvalidRegistry: ) Block: 3 Gas used: 447939 (3.73%)

Transaction sent: 0xfd5d681374a61c9bd3faab160ed8e178067f446e5defd4f5ead025170ab77d2
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 3
    GradientMarketMakerFactory.constructor confirmed Block: 4 Gas used: 5360588 (44.67%)
    GradientMarketMakerFactory deployed at: 0x6951b59d815043E3F842c1b026b0Fa888Cc2Dd85

Transaction sent: 0xa435993b2f2fb1d3287550dba6fc238c3794085270427a7f90342d2e9d0624c1
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 4
    MockEventAggregator.constructor confirmed Block: 5 Gas used: 451473 (3.76%)
    MockEventAggregator deployed at: 0xe0aA552A10d7EC8760Fc6c246D391E698a82dDf9

Transaction sent: 0xb7e625d259df2b4056775e29ffaca564b0ee6c59c2020b05d6e95b16439c4cc
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 0
    GradientMarketMakerFactory.setEventAggregator confirmed (reverted) Block: 6 Gas used: 22742 (0.19%)

Transaction sent: 0x6d55fedb01b9b5a7955dd98c2a103d6cd8f22313bcb5ce96a21025ebf52df6cf
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 5
    GradientMarketMakerFactory.setEventAggregator confirmed (reverted) Block: 7 Gas used: 22517 (0.19%)

Transaction sent: 0x49252a87f0c0f20f491e62dcf09c6b1ff01c7f4bbceeffb100658fa9a23049c
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 6
    GradientMarketMakerFactory.setEventAggregator confirmed Block: 8 Gas used: 30122 (0.25%)

tests/test_maker_factory.py::test_set_event_agg PASSED
tests/test_maker_factory.py::test_deploy_pool RUNNING
Transaction sent: 0x36ac3e58fab3b53c71a0a917152f2b7f3e81af07a66d1f3ca6d93ac8f834b208
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 7
    ERC20Mock.constructor confirmed Block: 9 Gas used: 523882 (4.37%)
    ERC20Mock deployed at: 0xcCB53c9429d32594F404d01fbe9E65ED1DCda8D9

Transaction sent: 0x965a84c8c2aa5504738a0f2e7aa1ccb810a730cb9a028dca44649b6826afa995
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 8
    ERC20Mock.constructor confirmed Block: 10 Gas used: 523882 (4.37%)
    ERC20Mock deployed at: 0x420b1099B9eF5bab0D92029594eF45E19A04A4A

Transaction sent: 0x0balc28e4cdelf1b3ef44084ad43aabde001d06ffed7f1b15bda34d9d6897179
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 9
    GradientRegistry.constructor confirmed Block: 11 Gas used: 1992969 (16.61%)
    GradientRegistry deployed at: 0xa3B53dDCd2E3fC28e8E130288F2aBD8d5EE37472

Transaction sent: 0x61bf0e6d7b7a3a97b2519dd54db7fdb1bf31f3ff8b415c94d284902b8171238
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 10
    MockEventAggregator.constructor confirmed Block: 12 Gas used: 451473 (3.76%)
    MockEventAggregator deployed at: 0xb6286fAFd0451320ad6A8143089b216C2152c025

Transaction sent: 0x749edbd5dd1476cc1a1a36a1bb76b863fb444f0089aec4bdbcce2ef9b12c84d29
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 11
    GradientMarketMakerFactory.constructor confirmed Block: 13 Gas used: 5360588 (44.67%)
    GradientMarketMakerFactory deployed at: 0x7a3d735ee6873f17Dbdcab1d518604928dc10d92

Transaction sent: 0x29ac634d451e28b045394cdd5977ccdd65d74247b3e714d0650d88dcfc436alf5
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 1
    GradientMarketMakerFactory.createPool confirmed (reverted) Block: 14 Gas used: 21652 (0.18%)

Transaction sent: 0xb828f6016a51e1693ba53ecc8ed92164cc05f225e871a2cc418795f6e89ed10b
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 12
    GradientRegistry.setTokenBlockStatus confirmed Block: 15 Gas used: 44060 (0.37%)

Transaction sent: 0x1411e778b28149d788f027431fbcd7ed36e17a02812d399200aa7a7ed00dcfc1
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 2
    GradientMarketMakerFactory.createPool confirmed (reverted) Block: 16 Gas used: 25112 (0.21%)

Transaction sent: 0x23272fba39672cb18c9a0f00b40b3e481d270c9efdf07cbaaada50c8bde0a60e
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 3
    GradientMarketMakerFactory.createPool confirmed Block: 17 Gas used: 4198329 (34.99%)

Transaction sent: 0x64ee7a2c85c3b7c4c3649598e6c441b7876da9f0501d02c15ffd2581c69341d8
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 4
    GradientMarketMakerFactory.createPool confirmed (reverted) Block: 18 Gas used: 22805 (0.19%)

Transaction sent: 0x2aa9b6f3284142a348973e93bb4d2a4ff7c419c56ee5506382a0c5e824bf7be1
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 13
    GradientRegistry.setTokenBlockStatus confirmed Block: 19 Gas used: 14524 (0.12%)

Transaction sent: 0x47a3bc2e0a363343238b0c4e6eca07c0e9df49694b16dd43b7a4183217fc5e92
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 5
    GradientMarketMakerFactory.createPool confirmed Block: 20 Gas used: 4183329 (34.86%)

tests/test_maker_factory.py::test_deploy_pool PASSED
```

```

tests/test_maker_factory.py::test_deploy_pool_liquidity RUNNING
Transaction sent: 0xc653fdce4edb0b683c5ffe8f546ba3ddc74f9e4de1ac638d4c91334735b6ea77
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 14
    UniswapV2Factory.constructor confirmed  Block: 21  Gas used: 2412742 (20.11%)
    UniswapV2Factory deployed at: 0xe65A7a341978d59d40d30FC23F5014FACB4f575A

Transaction sent: 0xfcclin7bb1174e2e1dc16beeab773e0c339c4d51ced533e52f74163bd374dba06
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 15
    WETH9.constructor confirmed  Block: 22  Gas used: 476546 (3.97%)
    WETH9 deployed at: 0x30375B532345B01cB8c2AD12541b09E9Aa53A93d

Transaction sent: 0x238992e8eb019de17fd5391a2cf379a055c484e3908b1dca2b7d02d6e5414c7d
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 16
    UniswapV2Router02.constructor confirmed  Block: 23  Gas used: 3895430 (32.46%)
    UniswapV2Router02 deployed at: 0x26f15335BB1C6a4C0B660eD694a0555A9F1cce3

Transaction sent: 0x43e8f49093637473190571bfe3803e0e11f255d05dcbd209bdbda656162b3895
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 17
    ERC20Mock.constructor confirmed  Block: 24  Gas used: 523930 (4.37%)
    ERC20Mock deployed at: 0xFbd588c72B438faD4Cf7cD879c8F730Faa213Da0

Transaction sent: 0x33094788f27725396fd4fdec6503e03ce451192958a2c064708fbe5d0f9fd0d4
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 18
    MockOrderbook.constructor confirmed  Block: 25  Gas used: 234574 (1.95%)
    MockOrderbook deployed at: 0xed00238F9a0F7b4d93842033cdF56cCB32C781c2

Transaction sent: 0xe601ecd1c21d2bf498ba839e095fe5620d1529fbfe3610c30da3a8797795fd75
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 19
    GradientRegistry.constructor confirmed  Block: 26  Gas used: 1992969 (16.61%)
    GradientRegistry deployed at: 0xDae02e4fE488952cFB8c95177154D188647a0146

Transaction sent: 0x3f3cc96711bcab94de2551da28c1c052b0692c4dc2b5f213f4811d7eb66e3b0f
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 20
    MockFallbackExecutor.constructor confirmed  Block: 27  Gas used: 793541 (6.61%)
    MockFallbackExecutor deployed at: 0xdCF93F11ef216cEC9C07fd31dD801c9b2b39Afb4

Transaction sent: 0xf2fdd656bda70fc7d98b5570f60247ba7fa1ac5d07aaaaea803edfe0ee23d1997
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 21
    MockEventAggregator.constructor confirmed  Block: 28  Gas used: 451473 (3.76%)
    MockEventAggregator deployed at: 0xBcb61491F1859f53438918F1A5aFCA542Af9D397

Transaction sent: 0xdf5921d24a8bfc2c43769a5c18bb41270dd9cb782b31f1642c9adf8df5c72915
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 22
    GradientMarketMakerFactory.constructor confirmed  Block: 29  Gas used: 5360588 (44.67%)
    GradientMarketMakerFactory deployed at: 0xD22363effe93190f82b52FC62B7Dcbc920eF658

Transaction sent: 0xb50d8a10cf2b20912e5211d50510fc82e1f7ee9d9455e006796dd2cbfc7734e
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 23
    GradientRegistry.setMainContracts confirmed  Block: 30  Gas used: 249856 (2.08%)

Transaction sent: 0xe4e2beb4c176b2601271b80a4ef492c0cc891e55b701554ed00ff421c6cad76
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 24
    ERC20Mock.constructor confirmed  Block: 31  Gas used: 523882 (4.37%)
    ERC20Mock deployed at: 0xf9c8Cf55f2E520B08d869df7bc76aa3d3ddDF913

Transaction sent: 0xa4355411677b65d19f65fb1e767e643c499a2a51dde18a3267fcf6aa3626795
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 25
    ERC20Mock.constructor confirmed  Block: 32  Gas used: 523882 (4.37%)
    ERC20Mock deployed at: 0x654f70d8442EA18904FA1AD79114f7250F7E9336

Transaction sent: 0xc73380633fa63ea4c96dd52c9799e134bdd1b5280b994e05b53fd74f0a3a15e6
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 26
    ERC20Mock.mint confirmed  Block: 33  Gas used: 65821 (0.55%)

Transaction sent: 0xf46b316654ea543722b626599672ea5a50cfdb8e0a4c36a0a74af5530b9d67c5
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 27
    ERC20Mock.mint confirmed  Block: 34  Gas used: 65821 (0.55%)

Transaction sent: 0xc9df3aa0ca3a20cdb55b307f4d7757f1d48dbdf5c9584clf5267fa26caec531
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 6
    ERC20Mock.approve confirmed  Block: 35  Gas used: 44271 (0.37%)

Transaction sent: 0xd4549d38362cf1210adcb2d1b8733cc7b3d742bd7760b18d2d5f053a7bece3a
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 7
    ERC20Mock.approve confirmed  Block: 36  Gas used: 44271 (0.37%)

Transaction sent: 0x4007e2f3622dd1a90e395dac0aa81aa8fd30c5a30ff26bdbcb7b5f284387f501
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 8
    GradientMarketMakerFactory.createPoolWithLiquidity confirmed (reverted)  Block: 37  Gas used: 22092 (0.18%)

Transaction sent: 0xb96dc47c948e724ae892003f84821f6439f6315d01eafe63afaafbfelbcd5aa
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 28
    GradientRegistry.setTokenBlockStatus confirmed  Block: 38  Gas used: 44060 (0.37%)

Transaction sent: 0xbc97e77eda98f0231e61a786f55f994f6832d3197e11304322a0574fbfd947fd
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 9
    GradientMarketMakerFactory.createPoolWithLiquidity confirmed (reverted)  Block: 39  Gas used: 25552 (0.21%)

Transaction sent: 0x17b1a365e8136455ff80fd73ca16d0a82ed7be59a3464977250a08ef7ca9b2e7
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 10
    GradientMarketMakerFactory.createPoolWithLiquidity confirmed (reverted)  Block: 40  Gas used: 25585 (0.21%)

```

```
Transaction sent: 0x3efa36dea5c258e4b047c68b7a694f2ebbd7d4f96f534db2261aef340ca05c13
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 29
    GradientRegistry.setTokenBlockStatus confirmed  Block: 41  Gas used: 14524 (0.12%)

Transaction sent: 0x4b7474417313fc9c2a1a8edc3d9aef48bb812b1f85b13b3fc91369fe69d2e670
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 11
    GradientMarketMakerFactory.createPool confirmed  Block: 42  Gas used: 4198329 (34.99%)

Transaction sent: 0x82837297c008e6fe1fab5f13d5a1315a85d2ff8fa7d45474f99f3c9cbf771bac
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 2
    GradientMarketMakerPoolV2.addTokenLiquidityForUser confirmed (reverted)  Block: 43  Gas used: 32493 (0.27%)

Transaction sent: 0xaafc566d7c9ea9b260ea46deacb1a220c2bd7fb20b455643049ac802ed185eaeb
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 3
    GradientMarketMakerPoolV2.addTokenLiquidityForUser confirmed (reverted)  Block: 44  Gas used: 44431 (0.37%)

Transaction sent: 0x64b2dbecc0dd384564d75b8a0f0d317622f03af398d736f1ef2419c3f5854657a
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 30
    UniswapV2Factory.createPair confirmed  Block: 45  Gas used: 2020041 (16.83%)

Transaction sent: 0x04161d8a88e307c91cfa303abffb83eb6f338a1077c6024968625eee0f0f5a25
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 12
    ERC20Mock.transfer confirmed  Block: 46  Gas used: 51120 (0.43%)

Transaction sent: 0x53253feb0e99d1e175cf0alde20a5a70187084337f367e556d583a7624de6319
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 4
    ERC20Mock.approve confirmed  Block: 47  Gas used: 44271 (0.37%)

Transaction sent: 0x0f26434dc6e7acf30c6e73906cab3ab4f1db2f863b9b6166b90e89492ca60c0f
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 5
    GradientMarketMakerPoolV2.addTokenLiquidityForUser confirmed  Block: 48  Gas used: 161532 (1.35%)

Transaction sent: 0xb9ceaaa2320b070f62c572b753d48fbfe0e53d68e107f87269840e1c0dc9ddd
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 6
    GradientMarketMakerPoolV2.addETHLiquidityForUser confirmed (reverted)  Block: 49  Gas used: 32168 (0.27%)

Transaction sent: 0x59ca201389d4ca5dd6c13a4eb4eb8d92abc1fe15478bb269ff053ff847c2f008
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 7
    GradientMarketMakerPoolV2.addETHLiquidityForUser confirmed  Block: 51  Gas used: 107805 (0.90%)

Transaction sent: 0xa9a4248281a6f8ab369baf3a4eb7a977a85a7b6ec4b2813525854f296d14e4ee
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 32
    UniswapV2Factory.createPair confirmed  Block: 52  Gas used: 2005041 (16.71%)

Transaction sent: 0xb5aa8099f6efd0b4f654a05c65cd4c63dad42c211dc667d9ea7582c0b9b2b89c
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 13
    GradientMarketMakerFactory.createPoolWithLiquidity confirmed  Block: 53  Gas used: 4449906 (37.08%)

tests/test_maker_factory.py::test_deploy_pool_liquidity PASSED
```

```
tests/test_maker_pool.py::test_add_liquidity RUNNING
Transaction sent: 0xa734d68f8b7af244568fd872fe14829079d42021cf28387837def4ece995c1fe
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 33
    UniswapV2Factory.constructor confirmed  Block: 54  Gas used: 2412742 (20.11%)
      UniswapV2Factory deployed at: 0x34b97ffa01dc0DC959c5f1176273D0de3be914C1

Transaction sent: 0x6955e97ee5b5231c706d5f41d891df7cd62668f9f7404c881f6a51075ec232b
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 34
    WETH9.constructor confirmed  Block: 55  Gas used: 476546 (3.97%)
      WETH9 deployed at: 0xc830Ad2FDfCC2f368fE5DeC93b1Dc72ecA8b3691

Transaction sent: 0x4a23783e923c98f18081b42e3921eacdce140cad9157af617f529a40efea0a4
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 35
    UniswapV2Router02.constructor confirmed  Block: 56  Gas used: 3895430 (32.46%)
      UniswapV2Router02 deployed at: 0xbc8eCcc89650c3E796e803CB009BF9b898CB359

Transaction sent: 0x1e828371b1call1fla8b9b0495d272d2263d8285500487d6a5488f162a4f447d3
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 36
    ERC20Mock.constructor confirmed  Block: 57  Gas used: 523882 (4.37%)
      ERC20Mock deployed at: 0x741e3E1f81041c62C2A97d0b6E567AcaB09A6232

Transaction sent: 0x1fb1272d700b1b18a12a22e147de738b2e7615110202117d9f82073e0f016fe9
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 37
    UniswapV2Factory.createPair confirmed  Block: 58  Gas used: 2020051 (16.83%)

Transaction sent: 0xe23c5a0aff26423ccbf3f2b68dad8060e042fefef88d499d13cefb3416b409a2
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 38
    ERC20Mock.constructor confirmed  Block: 59  Gas used: 523930 (4.37%)
      ERC20Mock deployed at: 0xFE0F4Cf81B5c0a6Fd65a610FD9488F33aE9095cB

Transaction sent: 0x1829b24b0488004b63b01b57f9f8efe90358abf6e5010166ad0f5b4847b0cd36
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 39
    MockOrderbook.constructor confirmed  Block: 60  Gas used: 234574 (1.95%)
      MockOrderbook deployed at: 0x4e3E7dc9D84dA7BE8f017f4C36153A61341736d4

Transaction sent: 0x9d2d8d869685efd220a7ebe7f0f018b0d7435ab476aae25f26a7f8ba823a2e6c
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 40
    GradientRegistry.constructor confirmed  Block: 61  Gas used: 1992969 (16.61%)
      GradientRegistry deployed at: 0x0AC45e945A008D3fc19da8f591be8601C1F63130

Transaction sent: 0xd8bf65f9be973f75b746adf763835b0daca36553ad07fbf849fcc27ba4ceee02
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 41
    MockFallbackExecutor.constructor confirmed  Block: 62  Gas used: 793529 (6.61%)
      MockFallbackExecutor deployed at: 0x5847798CE8c89e3Fff59AE5fA30BEC0d406b5687

Transaction sent: 0x3571c3b69201d9182f8c230cea423c8f3cb58e6f4d8d28201a4d5864ec7d6008
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 42
    MockEventAggregator.constructor confirmed  Block: 63  Gas used: 451473 (3.76%)
      MockEventAggregator deployed at: 0x0C60536783db9ED5A2B216970B10FF2243d317d0

Transaction sent: 0x198a9738edecbf07a4902eea82e8840a580dd2dbc743408e90cf66890c63d25
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 43
    GradientMarketMakerFactory.constructor confirmed  Block: 64  Gas used: 5360576 (44.67%)
      GradientMarketMakerFactory deployed at: 0x0e19E87b363DD7a3af5c45C95ab6885367251B94

Transaction sent: 0xa5b53191d0926c54f42759db9c01574f1b42486aacab8a59e62ef98d412cebe9
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 44
    GradientRegistry.setMainContracts confirmed  Block: 65  Gas used: 249868 (2.08%)

Transaction sent: 0xcafa447abc539790660762c8f0ca1d3ef0916c3331e69edb9ff8ef10b44e0ae3
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 45
    GradientMarketMakerFactory.createPool confirmed  Block: 66  Gas used: 4198329 (34.99%)

Transaction sent: 0xe135599b46d88f40dec203cb9b905582833d85627b563f4107e66b4bb48f7ca4
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 46
    ERC20Mock.mint confirmed  Block: 67  Gas used: 65821 (0.55%)

Transaction sent: 0xec205959db66bea92cb74b767df21a3a218465ead24784e317fed4c93abd22e
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 47
    ERC20Mock.mint confirmed  Block: 68  Gas used: 50809 (0.42%)

Transaction sent: 0x6c1468b66160c1fd379ba20621c323d971711607dd41a39bae3fdf85bad8b570
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 14
    ERC20Mock.approve confirmed  Block: 69  Gas used: 44271 (0.37%)

Transaction sent: 0x53da7805aa5286580bb9858e23b6a8e28a959657998673f26087e18308c13abb
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 0
    ERC20Mock.approve confirmed  Block: 70  Gas used: 44271 (0.37%)

Transaction sent: 0xf1663f5ae2e0b34f71d7c4e21e82bb8c80343622d1096b87abf6b632a32968ea
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 15
    GradientMarketMakerPoolV2.addTokenLiquidityWithProof confirmed (reverted)  Block: 71  Gas used: 33012 (0.28%)

Transaction sent: 0xb9d67c93e1d02cde926713af4daff69113fbfad7157470ad3ad35dd2e61cd3ba
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 16
    GradientMarketMakerPoolV2.addTokenLiquidityWithProof confirmed  Block: 72  Gas used: 180808 (1.51%)

Transaction sent: 0x0d5357719ce3dec0620a632b7f86708bde3544a866a996b8d905db36d119ef34
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 1
    GradientMarketMakerPoolV2.addTokenLiquidityWithProof confirmed  Block: 73  Gas used: 106398 (0.89%)
```

```
Transaction sent: 0xb976771630230e7f9e372b7bac90db1e499bc3ca02efad6a67acf7b5b9355e85
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 17
  GradientMarketMakerPoolV2.addETHLiquidityWithProof confirmed (reverted)  Block: 74  Gas used: 32820 (0.27%)

Transaction sent: 0xf9cb5273fa06e76841b5279d9b003d6944216e5ca0802b3c060984c6fbc64bf1
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 18
  GradientMarketMakerPoolV2.addETHLiquidityWithProof confirmed  Block: 75  Gas used: 113918 (0.95%)

tests/test_maker_pool.py::test_add_liquidity PASSED
```

```
tests/test_maker_pool.py::test_remove_liquidity RUNNING
Transaction sent: 0x820b9ba4457712d9b0a79c601f0ad2ff8f1c643de868e02adf6f9954727b7f9a
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 48
    UniswapV2Factory.constructor confirmed Block: 76 Gas used: 2412742 (20.11%)
    UniswapV2Factory deployed at: 0xb1B440724DCe2EE9779B58af841Ec59F545838B

Transaction sent: 0x4f50a7cd66cda90e64f586ff634dff3bb9d2a570132d65c818290507520b2113
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 49
    WETH9.constructor confirmed Block: 77 Gas used: 476546 (3.97%)
    WETH9 deployed at: 0xC6D563d5c2243b27e7294511063f563ED701EA2C

Transaction sent: 0x73f6b60115cab1680665eff019633b358a00ead676de2acff9745f23f0270baa
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 50
    UniswapV2Router02.constructor confirmed Block: 78 Gas used: 3895430 (32.46%)
    UniswapV2Router02 deployed at: 0xD537bF4b795b7D07Bd5F4bAf7017e3ce8360B1DE

Transaction sent: 0xf546d340c90c0e757fe7b641a002192767c80004f0ac4dae65f8260244a9d2fc
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 51
    ERC20Mock.constructor confirmed Block: 79 Gas used: 523882 (4.37%)
    ERC20Mock deployed at: 0x70bC6D873D110Da59a9c49E7485a27B0F605E5db

Transaction sent: 0x72fcc031ac98e73d75d175cfb185f526594698f0dabc1c0122cb6564939e790
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 52
    UniswapV2Factory.createPair confirmed Block: 80 Gas used: 2020051 (16.83%)

Transaction sent: 0x9773bebe67059354b70b55097386a30f1f3b5961ad9c264cad9f703c3e393e1d
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 53
    ERC20Mock.constructor confirmed Block: 81 Gas used: 523930 (4.37%)
    ERC20Mock deployed at: 0xb291024F4b2E4C3767A96aB44566a8e93cC828C5

Transaction sent: 0x0b36458994a878e51d1eb07565dee0bba978592e96f39d771a54e7771de9fcfa
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 54
    MockOrderbook.constructor confirmed Block: 82 Gas used: 234574 (1.95%)
    MockOrderbook deployed at: 0xEfE66132727f3831AB4E020357B587615076Df6A

Transaction sent: 0xc6f0c7b650348929c8862696c52bcd5bd29fbfd6d616258c3a9130b6c5741d675
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 55
    GradientRegistry.constructor confirmed Block: 83 Gas used: 1992969 (16.61%)
    GradientRegistry deployed at: 0xEFDE7eC452f9A203559896F64267809E248D044f

Transaction sent: 0xb3e3c3cb850ae963bed9c219e4495ff7d0a2e7b6a5701c75ac91c3c790211253
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 56
    MockFallbackExecutor.constructor confirmed Block: 84 Gas used: 793541 (6.61%)
    MockFallbackExecutor deployed at: 0x93F6bA0E6DcB1Ec0F1aBD80Be31e025d6e64Ad97

Transaction sent: 0x958b9812d5572e0849ba18af2c6d50ebb54806a9918fe30b3a432c7e3a64932b
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 57
    MockEventAggregator.constructor confirmed Block: 85 Gas used: 451473 (3.76%)
    MockEventAggregator deployed at: 0x342216ff4b4D6899DDE6535D6E50b6Db71654bc4

Transaction sent: 0x29584464292a4cb75f479926b4c40719eb4f45f6ef870ad3d48c3b7879228ae5
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 58
    GradientMarketMakerFactory.constructor confirmed Block: 86 Gas used: 5360588 (44.67%)
    GradientMarketMakerFactory deployed at: 0xa5659Fa683b71C907cd346397c694b93aFF40b51

Transaction sent: 0xd26ae348fe8a8c20b55231001a634167da0a5b7d19a23c3a4e03df9f439855d0
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 59
    GradientRegistry.setMainContracts confirmed Block: 87 Gas used: 249868 (2.08%)

Transaction sent: 0x1efa2e6d42a10ea5f41e27336b5461ee841f2a263ac34129cbc9a4f64cde99d5
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 60
    GradientMarketMakerFactory.createPool confirmed Block: 88 Gas used: 4198329 (34.99%)

Transaction sent: 0xd33ebcbe34f3b4baef1f5cae4f012c5677ae04907c344d73d5c48a802ddefe7a
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 61
    ERC20Mock.mint confirmed Block: 89 Gas used: 65821 (0.55%)

Transaction sent: 0x27cffc96b90648b38509afbae0662fb7104aba48dae67c0f752d47d9cdc19c62
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 62
    ERC20Mock.mint confirmed Block: 90 Gas used: 50809 (0.42%)

Transaction sent: 0x505b8f6c7e611bf91dal2d187da43b260ab6d0cfbe2961412fbe5fba059141
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 19
    ERC20Mock.approve confirmed Block: 91 Gas used: 44271 (0.37%)

Transaction sent: 0x762d0cd7edf769042c1bd330c7599dc6b094e1c23a41f875877d887601ce6a5f
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 2
    ERC20Mock.approve confirmed Block: 92 Gas used: 44271 (0.37%)

Transaction sent: 0xe43e2eaec2c1d105c26c348698d1ee04ce1a265378aa31f4246e86c93647a719
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 20
    GradientMarketMakerPoolV2.addTokenLiquidityWithProof confirmed Block: 93 Gas used: 180808 (1.51%)

Transaction sent: 0xb47355e4f2b95e536b15a27f76b4308ad7fcdeb75b7f9907608c534f75deaf2
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 3
    GradientMarketMakerPoolV2.addTokenLiquidityWithProof confirmed Block: 94 Gas used: 106398 (0.89%)

Transaction sent: 0xb09e03c42ff6da21089efda807e0e48a9498999c9893a0a31424cf0a2d87a87b
    Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 21
    GradientMarketMakerPoolV2.removeTokenLiquidityWithProof confirmed (reverted) Block: 95 Gas used: 84264 (0.70%)
```

```
Transaction sent: 0x52b98b5c824f22351dbe4edb3172fed1030ebf5aaaf9e9beb1a723d4657802a
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 22
GradientMarketMakerPoolV2.removeTokenLiquidityWithProof confirmed (reverted) Block: 96 Gas used: 32379 (0.27%)

Transaction sent: 0x7009970ea4cf3fbedd64a87690111398afba1edc2b42f38d28a8bf41692307a3
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 23
GradientMarketMakerPoolV2.removeTokenLiquidityWithProof confirmed Block: 97 Gas used: 105613 (0.88%)

Transaction sent: 0xc5bf9705fce1ec155e2ea13ef6fd60e26a4f51ad02d4d09d06586debc77e1ac7
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 24
GradientMarketMakerPoolV2.removeETHLiquidityWithProof confirmed (reverted) Block: 98 Gas used: 38835 (0.32%)

Transaction sent: 0x0206988cf955637f83bb1609f89dd0d6e5b12b3e0222649a22facabe9ec37cb1
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 25
GradientMarketMakerPoolV2.removeETHLiquidityWithProof confirmed (reverted) Block: 99 Gas used: 32467 (0.27%)

Transaction sent: 0x612dc8fc53e3848853aea66576446e39644eca666deb5fb692b32c73c667fe75
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 26
GradientMarketMakerPoolV2.addETHLiquidityWithProof confirmed Block: 100 Gas used: 113918 (0.95%)

Transaction sent: 0xfe951732e7b1276861b77e2df5b02b99b99ac7dc52e6c7681d90fae095629d68
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 4
GradientMarketMakerPoolV2.addETHLiquidityWithProof confirmed Block: 101 Gas used: 83918 (0.70%)

Transaction sent: 0xb38327a20745a5c387a80d7fc1f8145933c663461810dd809463680c6b8f3a1e
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 27
GradientMarketMakerPoolV2.removeETHLiquidityWithProof confirmed Block: 102 Gas used: 96609 (0.81%)

tests/test_maker_pool.py::test_remove_liquidity PASSED
```

```
tests/test_maker_pool.py::test_buy_sell_order RUNNING
Transaction sent: 0xd9c6ebd54f8113798fd0e12a8faed14cf082b93c84eacd9e788d30ce6c963d8c
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 63
  UniswapV2Factory.constructor confirmed  Block: 103  Gas used: 2412742 (20.11%)
  UniswapV2Factory deployed at: 0x0764A8da0FCAA127b31E13b79E9a273Fb5B2bF26

Transaction sent: 0x2cc6423c40dc6aa11cc1fc4772e2cd806d4e5c27185331f3f2ecc29d772e74bb
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 64
  WETH9.constructor confirmed  Block: 104  Gas used: 476546 (3.97%)
  WETH9 deployed at: 0x2acC1AbFdC76E48e49c91f55512066B4DD6555Dd

Transaction sent: 0x42dddfbfec34978cad050c84f955211be8d6636184791e9e33a25e8d1352361b
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 65
  UniswapV2Router02.constructor confirmed  Block: 105  Gas used: 3895430 (32.46%)
  UniswapV2Router02 deployed at: 0xFd87ceEa79b211268Fb182a48E43644543c08709

Transaction sent: 0x6dd185deac5d4e8ec4e25129f605f004c454c050b5206db46b8c817fd9e2bdf0
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 66
  ERC20Mock.constructor confirmed  Block: 106  Gas used: 523882 (4.37%)
  ERC20Mock deployed at: 0x7e4ED6273177126c48503850a526845bed16E747

Transaction sent: 0x181bdb435afcb0cc3900873135fd11fa8b0960a0a176ebdc73a0eba68765a89
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 67
  UniswapV2Factory.createPair confirmed  Block: 107  Gas used: 2020041 (16.83%)

Transaction sent: 0x7297ed05ce5289850582c9714a583eb0919ae591c489a29c74efde59dc749b09
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 68
  ERC20Mock.constructor confirmed  Block: 108  Gas used: 523930 (4.37%)
  ERC20Mock deployed at: 0x6a21483D01cFb6Faf0D5346C1e781337425D713B

Transaction sent: 0xb56118e6d271ebcb2e90e6ff99156a41d98198d63b9407c3856d1e7320a4de68
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 69
  MockOrderbook.constructor confirmed  Block: 109  Gas used: 234574 (1.95%)
  MockOrderbook deployed at: 0x7Bcb5DCBE1d42458D8C5b2397C68d18198E8066

Transaction sent: 0xccf272f16ef9ef58161764554d311b7538bc23d3517c440ba8884281be4d8d06
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 70
  GradientRegistry.constructor confirmed  Block: 110  Gas used: 1992969 (16.61%)
  GradientRegistry deployed at: 0xaE4059E2E23958f627eF75d2467343F4d8838709

Transaction sent: 0xe86f991667d1a12c5ebb4fadcl14b3fe157fba4ee7bef098adf49505ccb9014
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 71
  MockFallbackExecutor.constructor confirmed  Block: 111  Gas used: 793541 (6.61%)
  MockFallbackExecutor deployed at: 0x47a27C9881B5C5eC78c38FBEE7DD0A5eb68633C

Transaction sent: 0x246d6541401dfa3276789a08937fff248c09dd5af9f05447027c89f23ba3761
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 72
  MockEventAggregator.constructor confirmed  Block: 112  Gas used: 451473 (3.76%)
  MockEventAggregator deployed at: 0x63Cf24e22bbFAA28316Cca9b179eEC7810b8037B

Transaction sent: 0x5da8dc41790ad34a05084064b1860eaacca41beae4076cdd6232f1e41e37cd695
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 73
  GradientMarketMakerFactory.constructor confirmed  Block: 113  Gas used: 5360588 (44.67%)
  GradientMarketMakerFactory deployed at: 0x8dB41305cA5f2852135dC2B969ba6084A32D4700

Transaction sent: 0xf4dfc8bb63f6ea767267c8f8c38bc3a33ba922fa302ee78cb1e90e1611f293b2
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 74
  GradientRegistry.setMainContracts confirmed  Block: 114  Gas used: 249856 (2.08%)

Transaction sent: 0x15bc1e99d0cc9c73b8157d2cc1a7195135d91a845353941c833f0db9b3ae14cc
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 75
  GradientMarketMakerFactory.createPool confirmed  Block: 115  Gas used: 4198329 (34.99%)

Transaction sent: 0xc754b0fa03defe90f01604269d33fa6545b8690578f714ea016e4a850a22f88
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 76
  ERC20Mock.mint confirmed  Block: 116  Gas used: 65821 (0.55%)

Transaction sent: 0x25f32e5ecae95230c9214600aaec0e75aa4905f7c1fe1772939a48748087a0a9
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 77
  ERC20Mock.mint confirmed  Block: 117  Gas used: 50809 (0.42%)

Transaction sent: 0xba76b70a6cdadcdcae39d81e2fbfc571acd330336d079bd9d684efd522ae66451
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 28
  ERC20Mock.approve confirmed  Block: 118  Gas used: 44271 (0.37%)

Transaction sent: 0x7df9f180c446bae2b96b8e3e3fe5d4b4c57ee7226fa9754f38575de5486a9fad
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 5
  ERC20Mock.approve confirmed  Block: 119  Gas used: 44271 (0.37%)

Transaction sent: 0xa53b2cba33f1ebf833bfc40b78d970809f3e5e9a2042f9ec43ce3152800820a8
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 6
  GradientMarketMakerPoolV2.addTokenLiquidityWithProof confirmed  Block: 120  Gas used: 180808 (1.51%)

Transaction sent: 0xe41d96866385152a8871ae8267de350a5fe8bfff6fc7875074750913ef078a06
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 29
  GradientMarketMakerPoolV2.addETHLiquidityWithProof confirmed  Block: 121  Gas used: 112214 (0.94%)

Transaction sent: 0x0307e86e6a38883bd67ff541635f0d140b449ff6b9cf38ebb75802c60e06582a
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 78
  GradientMarketMakerPoolV2.executeBuyOrder confirmed (reverted)  Block: 122  Gas used: 29029 (0.24%)
```

```
Transaction sent: 0x03a506fc3e8548da70ce2cf43bc993e76678af0e273a48cb2225031121eae81d
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 1
  GradientMarketMakerPoolV2.executeBuyOrder confirmed (reverted)  Block: 123  Gas used: 29016 (0.24%)

Transaction sent: 0xc5eefcacc9d7b69e021c6e0651066b98e68956d1e878d43bf2ad5f4ca970b912
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 2
  GradientMarketMakerPoolV2.executeBuyOrder confirmed (reverted)  Block: 124  Gas used: 29027 (0.24%)

Transaction sent: 0xebf7452547a746ae55bc67c57289b7f418e0794655973a8d0ea96ed1d7480dd2
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 3
  GradientMarketMakerPoolV2.executeBuyOrder confirmed (reverted)  Block: 125  Gas used: 29121 (0.24%)

Transaction sent: 0x9889b8e2b57c01e35caaae30d4c17c8126290b3c124bf0637d71325554c763ac
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 4
  GradientMarketMakerPoolV2.executeBuyOrder confirmed (reverted)  Block: 127  Gas used: 29947 (0.25%)

Transaction sent: 0x38887466900c917935671342d33f654ebe5ef3a8ab9299f328eb113d799d1d8f
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 5
  GradientMarketMakerPoolV2.executeBuyOrder confirmed  Block: 128  Gas used: 109682 (0.91%)

Transaction sent: 0xa74b664379903134f02dc06e266b89cd67fd9da97f35396ee7a932ef35b1d430
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 80
  GradientMarketMakerPoolV2.executeSellOrder confirmed (reverted)  Block: 129  Gas used: 29141 (0.24%)

Transaction sent: 0xf9657a2158a4297ef16c57c9f0df675aee9823f7b93250da7535093fdd209537
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 6
  GradientMarketMakerPoolV2.executeSellOrder confirmed (reverted)  Block: 130  Gas used: 29128 (0.24%)

Transaction sent: 0xbeaecd79563c2e9dcc076a95ece21c367e17d67dcfba89215df4e7685fe82ae
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 7
  GradientMarketMakerPoolV2.executeSellOrder confirmed (reverted)  Block: 131  Gas used: 29139 (0.24%)

Transaction sent: 0xbc91c0cff3acba79e10399c70584fae25f0646c3cb76d91b5576e16b614009e
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 8
  GradientMarketMakerPoolV2.executeSellOrder confirmed (reverted)  Block: 132  Gas used: 30049 (0.25%)

Transaction sent: 0xc0eaf2761d129d11a99fa84ace3f43b7b0727ad046010af2ba153043e505f12f
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 9
  ERC20Mock.approve confirmed  Block: 133  Gas used: 44271 (0.37%)

Transaction sent: 0xe4f5b0a57f006a0421354ebd4db04dd304de93a1e537dc29f1baeeb0ae579677
  Gas price: 0.0 gwei  Gas limit: 12000000  Nonce: 10
  GradientMarketMakerPoolV2.executeSellOrder confirmed  Block: 134  Gas used: 93962 (0.78%)

tests/test_maker_pool.py::test_buy_sell_order PASSED
```

```
tests/test_maker_pool.py::test_set_min_liquidity RUNNING
Transaction sent: 0xa2e8b87fc47ece25cc6fc0c7c244554dd1f696a18562b43802f9169eb14a7e66
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 81
    UniswapV2Factory.constructor confirmed  Block: 135  Gas used: 2412742 (20.11%)
    UniswapV2Factory deployed at: 0x741D86951a8b99159be52644e60de7826033Ee44

Transaction sent: 0x63280274e830398540bf99aae8945fc16326631947af16cc86674e0b8493c928
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 82
    WETH9.constructor confirmed  Block: 136  Gas used: 476546 (3.97%)
    WETH9 deployed at: 0x953d9B708E446dcC9257d900C7cFc34Db2969A57

Transaction sent: 0xc26276b6fb4cf37fb7c47f9cacead68ddefe5f52805416287fdc8fb3863ba4fa
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 83
    UniswapV2Router02.constructor confirmed  Block: 137  Gas used: 3895418 (32.46%)
    UniswapV2Router02 deployed at: 0x4f53C468a3AaCle528C2AB15f18D777D083d642E

Transaction sent: 0xd54d94d9982b29f7a174282f61f4bb0dd77cfee7bc322309acdae6f2beb4592d
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 84
    ERC20Mock.constructor confirmed  Block: 138  Gas used: 523882 (4.37%)
    ERC20Mock deployed at: 0x5F589a452BEF32ae54Fbc46968505867cEFbC537

Transaction sent: 0xc77a9e0903759409134eafbb9cb93295ca6929135369b85c5dc1081ebd89e40
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 85
    ERC20Mock.constructor confirmed  Block: 139  Gas used: 523930 (4.37%)
    ERC20Mock deployed at: 0x54BA761038f133140659d8a5a4375b479D7c5582

Transaction sent: 0xc40d826627ed9863fb7d4e741874a8c72294b843a9d137d04ca18ca0e7a38be4
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 86
    MockOrderbook.constructor confirmed  Block: 140  Gas used: 234574 (1.95%)
    MockOrderbook deployed at: 0x5F6F5B3b1b4d2B448bcbffA9b92168469ab2e142

Transaction sent: 0x27f2465196ddd3fb265eeb5a0e959d5b43fe012b5bb8d6efdf5e6a627b6861d
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 87
    GradientRegistry.constructor confirmed  Block: 141  Gas used: 1992969 (16.61%)
    GradientRegistry deployed at: 0x6A547C98A4363b71F74472aD25595F4677a8b94B

Transaction sent: 0x1c9245a0d0d3939294fc284fb1356e22175bc49beda3a5cf738e63ffa7bdfe88
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 88
    MockFallbackExecutor.constructor confirmed  Block: 142  Gas used: 793541 (6.61%)
    MockFallbackExecutor deployed at: 0x1c7f458653173531D50a0C6c85a8e390D09E5D4E

Transaction sent: 0x30dff6e098d7a3924189cac3380ce08ae100d31dd5f162f4408d3066dad1d446
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 89
    MockEventAggregator.constructor confirmed  Block: 143  Gas used: 451473 (3.76%)
    MockEventAggregator deployed at: 0x12429A7C54a315b8325Ae6105417b04BEE258a27

Transaction sent: 0x678892e7640bddc0ac5eb85d0af5c52b3fcfc9206f9329b71f1eeeabb3c48605
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 90
    GradientMarketMakerFactory.constructor confirmed  Block: 144  Gas used: 5360588 (44.67%)
    GradientMarketMakerFactory deployed at: 0xe372E377902f1872Aaa458989B229974896022D2

Transaction sent: 0xb5d0a30c89d7ce30950cb4228c3ec7516bcd4a5623df0dd9db6d694fde98c720
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 91
    GradientRegistry.setMainContracts confirmed  Block: 145  Gas used: 249868 (2.08%)

Transaction sent: 0xffe9e5436fe54476d27fc848936ceffab7a2950fcda73f6231fa75cd6b44d6
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 92
    GradientMarketMakerFactory.createPool confirmed  Block: 146  Gas used: 4198329 (34.99%)

Transaction sent: 0x5583a2cb4e3491b67769e48e368d139494ef6c674c38c7d6b3d1fa784ed13eb4
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 30
    GradientMarketMakerPoolV2.setMinLiquidity confirmed (reverted)  Block: 147  Gas used: 22510 (0.19%)

Transaction sent: 0xa84ac5e62432794eeae733645e1f67bf712cd65e0124d5fd7bbe1d69ddf8d0f6
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 93
    GradientMarketMakerPoolV2.setMinLiquidity confirmed (reverted)  Block: 148  Gas used: 22453 (0.19%)

Transaction sent: 0x4fcc70482fae054b4561ff66f69b354b92b1541ed11d658a40e6f76163d6330
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 94
    GradientMarketMakerPoolV2.setMinLiquidity confirmed  Block: 149  Gas used: 28541 (0.24%)

tests/test_maker_pool.py::test_set_min_liquidity PASSED
```

```
tests/test_maker_pool.py::test_set_min_token_liquidity RUNNING
Transaction sent: 0x6b35f50104e793cbc4a0db2ae6c1f81c925e49df8b5ce3a5e174f0b4990c5c58
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 95
    UniswapV2Factory.constructor confirmed  Block: 150  Gas used: 2412742 (20.11%)
    UniswapV2Factory deployed at: 0xE67cf20A24c85f0417070668123102247a6fcfcC1

Transaction sent: 0xde195f2e1b552595c5fde3914ffab84ad4ab0505e6c605830677e16ec7150353
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 96
    WETH9.constructor confirmed  Block: 151  Gas used: 476546 (3.97%)
    WETH9 deployed at: 0xFE78A958963c513B22337B669EdaF764477abd6E

Transaction sent: 0x0ff79479fa28119779239dff2f0134d4053acaee1c57af53d56dcda65c68e824
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 97
    UniswapV2Router02.constructor confirmed  Block: 152  Gas used: 3895430 (32.46%)
    UniswapV2Router02 deployed at: 0x683554B7d4a805c1B0A6Da16bbb85707210Cce0E

Transaction sent: 0xd92510926149a9bdf6e03a881eb047cfde8957a8f852dae3eb54d525367b34de
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 98
    ERC20Mock.constructor confirmed  Block: 153  Gas used: 523882 (4.37%)
    ERC20Mock deployed at: 0xc44163a3C9a048c7EdC2A74299146173eaF436AF

Transaction sent: 0x393887d0ad178308c5fd1e08ef93f658c21263e5bf079adea9a2709c1f7ca366
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 99
    ERC20Mock.constructor confirmed  Block: 154  Gas used: 523930 (4.37%)
    ERC20Mock deployed at: 0xa2c283F11108596AB76e47af4e487Ec5D9d6c5BF

Transaction sent: 0xe1c6eaed31ea0955c050b33191fb4f1a5f9cec38017be137def1124e7395afa0
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 100
    MockOrderbook.constructor confirmed  Block: 155  Gas used: 234574 (1.95%)
    MockOrderbook deployed at: 0x2711D43E99752B654E6925861Aec015E1F635cBb

Transaction sent: 0xe48b66023abed39d99a5eb027543d69a2e585819e8df932b659bd8ea63c2011b
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 101
    GradientRegistry.constructor confirmed  Block: 156  Gas used: 1992969 (16.61%)
    GradientRegistry deployed at: 0x2847751f6a9902c2AB6e3C319901aCaC184f81b6

Transaction sent: 0x140989964c41d14e4b6aa127454709e13741f22f1b7154362abf3e87916f6371
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 102
    MockFallbackExecutor.constructor confirmed  Block: 157  Gas used: 793541 (6.61%)
    MockFallbackExecutor deployed at: 0xc9D90614566289F796Eb9272a5ed37B8e311eC38

Transaction sent: 0xbcea058912f33e777b0981c3ae2d240a34f2a80e3fc5ae18e9a1db0f70062ce7
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 103
    MockEventAggregator.constructor confirmed  Block: 158  Gas used: 451473 (3.76%)
    MockEventAggregator deployed at: 0xb836484F58733Eb4254dbf20B89a313D34BBEf5b

Transaction sent: 0xa7dd63dea4bf8070b925bb208571870e7137dff4d8335e59f36634fc7282cd0
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 104
    GradientMarketMakerFactory.constructor confirmed  Block: 159  Gas used: 5360588 (44.67%)
    GradientMarketMakerFactory deployed at: 0xA671b1854BaF6e1c29Bb5D950c047E9964721076

Transaction sent: 0x872cd7a365c2d7d07746e8830b045f597de211ff69924d4280d549ce7df2a25
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 105
    GradientRegistry.setMainContracts confirmed  Block: 160  Gas used: 249868 (2.08%)

Transaction sent: 0x432769802ef625700d606cbdef4fb1923ada34b0b29498dfe8dd1f4667fbe6c
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 106
    GradientMarketMakerFactory.createPool confirmed  Block: 161  Gas used: 4198329 (34.99%)

Transaction sent: 0xe14cc859854b01146d4909633a978123ef456368d4205e98a47022390f705ab2
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 31
    GradientMarketPoolV2.setMinTokenLiquidity confirmed (reverted)  Block: 162  Gas used: 22543 (0.19%)

Transaction sent: 0xbe8f092748ea0fee05584d356d43dbdb2900ad23b4bedb1d23da27ecd0643ae7
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 107
    GradientMarketPoolV2.setMinTokenLiquidity confirmed (reverted)  Block: 163  Gas used: 22474 (0.19%)

Transaction sent: 0x1ad37342010fcab0b93e0177bb354b8b2ae6086ad337bc0ce8f02dcdf826fc02
  Gas price: 0.0 gwei  Gas limit: 12000000 Nonce: 108
    GradientMarketPoolV2.setMinTokenLiquidity confirmed  Block: 164  Gas used: 28585 (0.24%)

tests/test_maker_pool.py::test_set_min_token_liquidity PASSED
```

Annexes

Testing code:

Factory:

```
from brownie import (
    reverts,
    accounts,
    GradientMarketMakerPoolV2
)

from scripts.helpful_scripts import (
    ZERO_ADDRESS,
    DAY_TIMESTAMP,
    get_account,
)

from scripts.deploy import (
    deploy_erc,
    deploy_factory,
    deploy_weth,
    deploy_router,
    deploy_registry,
    deploy_event_aggregator,
    deploy_fallback_executor,
    deploy_orderbook,
    deploy_maker_factory
)

def test_set_event_agg(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)
```

```

registry = deploy_registry(owner)
aggregator = deploy_event_aggregator(owner)

with reverts("InvalidRegistry: "):
    deploy_maker_factory(owner, ZERO_ADDRESS, aggregator.address)

maker_factory = deploy_maker_factory(owner, registry.address,
aggregator.address)
new_aggregator = deploy_event_aggregator(owner)
with reverts():
    maker_factory.setEventAggregator(new_aggregator.address, {"from": other})
with reverts("InvalidEventAggregator: "):
    maker_factory.setEventAggregator(ZERO_ADDRESS, {"from": owner})

assert maker_factory.eventAggregator() == aggregator.address
tx = maker_factory.setEventAggregator(new_aggregator.address, {"from": owner})
assert tx.events['EventAggregatorUpdated'][0]['oldEventAggregator'] ==
aggregator.address
assert tx.events['EventAggregatorUpdated'][0]['newEventAggregator'] ==
new_aggregator.address
assert maker_factory.eventAggregator() == new_aggregator.address

def test_deploy_pool(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)

    token1 = deploy_erc(owner, "token1", "token1")
    token2 = deploy_erc(owner, "token2", "token2")

    registry = deploy_registry(owner)
    aggregator = deploy_event_aggregator(owner)
    maker_factory = deploy_maker_factory(owner, registry.address,

```

```
aggregator.address)

with reverts("InvalidTokenAddress: "):
    maker_factory.createPool(ZERO_ADDRESS, {"from": other})
# Block a token
registry.setTokenBlockStatus(token1.address, True, {"from": owner})
with reverts("TokenBlocked: "):
    maker_factory.createPool(token1.address, {"from": other})

tx = maker_factory.createPool(token2.address, {"from": other})
assert tx.events['PoolCreated'][0]['token'] == token2.address
assert tx.events['PoolCreated'][0]['poolIndex'] == 0

with reverts("PoolAlreadyExists: "):
    maker_factory.createPool(token2.address, {"from": other})

# Unblock a token
registry.setTokenBlockStatus(token1.address, False, {"from": owner})

tx = maker_factory.createPool(token1.address, {"from": other})
assert tx.events['PoolCreated'][0]['token'] == token1.address
assert tx.events['PoolCreated'][0]['poolIndex'] == 1

def test_deploy_pool_liquidity(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)

    # Deploy uniswap
    factory = deploy_factory(owner, owner)
    weth = deploy_weth(owner)
    router = deploy_router(owner, factory.address, weth.address)
```

```

# Deploy registry

gradient_token = deploy_erc(owner, "gradient", "gradient")
orderbook = deploy_orderbook(owner)
registry = deploy_registry(owner)
fallback_executor = deploy_fallback_executor(owner, registry.address)

aggregator = deploy_event_aggregator(owner)
maker_factory = deploy_maker_factory(owner, registry.address,
aggregator.address)

registry.setMainContracts(
    maker_factory.address,
    gradient_token.address,
    orderbook.address,
    fallback_executor.address,
    router.address,
    {"from": owner}
)

token1 = deploy_erc(owner, "token1", "token1")
token2 = deploy_erc(owner, "token2", "token2")
token1.mint(other, 10e18)
token2.mint(other, 10e18)
token1.approve(maker_factory.address, 10e18, {"from": other})
token2.approve(maker_factory.address, 10e18, {"from": other})

with reverts("InvalidTokenAddress: "):
    maker_factory.createPoolWithLiquidity(ZERO_ADDRESS, 1e18, 1e17, {"from": other})

    # Block a token
    registry.setTokenBlockStatus(token1.address, True, {"from": owner})
    with reverts("TokenBlocked: "):
        maker_factory.createPoolWithLiquidity(token1.address, 1e18, 1e17, {"from": other})

    with reverts("EthAmountMismatch: "):

```

```

        maker_factory.createPoolWithLiquidity(token2.address, 1e18, 1e17, {"from": other})

        registry.setTokenBlockStatus(token1.address, False, {"from": owner})

        maker_factory.createPool(token1.address, {"from": other})
        pool_address = maker_factory.getPool(token1.address)
        pool = GradientMarketMakerPoolV2.at(pool_address)

    with reverts("TokenAmountBelowMinimum: "):
        pool.addTokenLiquidityForUser(other, 1e18, {"from": maker_factory.address})
    with reverts("PairDoesNotExist: "):
        pool.addTokenLiquidityForUser(other, 2e18, {"from": maker_factory.address})

    # Create pair
    factory.createPair(weth.address, token1.address)
    token1.transfer(maker_factory.address, 2e18, {"from": other})
    token1.approve(pool, 10e18, {"from": maker_factory.address})
    tx = pool.addTokenLiquidityForUser(other, 2e18, {"from": maker_factory.address})
    assert tx.events['TokenLiquidityDeposited'][0]['user'] == other
    assert tx.events['TokenLiquidityDeposited'][0]['token'] == token1.address
    assert tx.events['TokenLiquidityDeposited'][0]['tokenAmount'] == 2e18

    with reverts("ETHAmountBelowMinimum: "):
        pool.addETHLiquidityForUser(extra, {"from": maker_factory.address})

    owner.transfer(maker_factory.address, "10 ether")
    tx = pool.addETHLiquidityForUser(extra, {"from": maker_factory.address, "value": 1e17})
    assert tx.events['ETHLiquidityDeposited'][0]['user'] == extra
    assert tx.events['ETHLiquidityDeposited'][0]['token'] == token1.address
    assert tx.events['ETHLiquidityDeposited'][0]['ethAmount'] == 1e17

    factory.createPair(weth.address, token2.address)
    tx = maker_factory.createPoolWithLiquidity(token2.address, 5e17, 2e18, {"from": other, "value": 5e17})
    assert tx.events['TokenLiquidityDeposited'][0]['user'] == other

```

```

assert tx.events['TokenLiquidityDeposited'][0]['token'] == token2.address
assert tx.events['TokenLiquidityDeposited'][0]['tokenAmount'] == 2e18
assert tx.events['ETHLiquidityDeposited'][0]['user'] == other
assert tx.events['ETHLiquidityDeposited'][0]['token'] == token2.address
assert tx.events['ETHLiquidityDeposited'][0]['ethAmount'] == 5e17

```

Pool:

```

from brownie import (
    reverts,
    accounts,
    GradientMarketMakerPoolV2
)

from scripts.helpful_scripts import (
    ZERO_ADDRESS,
    DAY_TIMESTAMP,
    get_account,
)

from scripts.deploy import (
    deploy_erc,
    deploy_factory,
    deploy_weth,
    deploy_router,
    deploy_registry,
    deploy_event_aggregator,
    deploy_fallback_executor,
    deploy_orderbook,
    deploy_maker_factory
)

def test_add_liquidity(only_local):
    # Arrange
    owner = get_account(0)

```

```

other = get_account(1)
extra = get_account(2)

# Deploy uniswap
factory = deploy_factory(owner, owner)
weth = deploy_weth(owner)
router = deploy_router(owner, factory.address, weth.address)

token1 = deploy_erc(owner, "token1", "token1")
factory.createPair(weth.address, token1.address)
maker_factory, pool, _ = deploy_pool(owner, token1.address, router.address)

# mint some tokens
token1.mint(other, 10e18)
token1.mint(extra, 10e18)
token1.approve(pool, 10e18, {"from": other})
token1.approve(pool, 10e18, {"from": extra})

with reverts("TokenAmountBelowMinimum: "):
    pool.addTokenLiquidityWithProof(
        1e18,
        [],
        # Empty proof
        0, 0, 0, 0, # Dummy position values
        {"from": other}
    )

tx = pool.addTokenLiquidityWithProof(
    2e18,
    [],
    # Empty proof
    0, 0, 0, 0, # Dummy position values
    {"from": other}
)
assert tx.events['TokenLiquidityDeposited'][0]['user'] == other
assert tx.events['TokenLiquidityDeposited'][0]['token'] == token1.address

```

```

assert tx.events['TokenLiquidityDeposited'][0]['tokenAmount'] == 2e18

tx = pool.addTokenLiquidityWithProof(
    2e18,
    [],
        # Empty proof
    0, 0, 0, 0, # Dummy position values
    {"from": extra}
)

assert tx.events['TokenLiquidityDeposited'][0]['user'] == extra
assert tx.events['TokenLiquidityDeposited'][0]['token'] == token1.address
assert tx.events['TokenLiquidityDeposited'][0]['tokenAmount'] == 2e18

with reverts("ETHAmountBelowMinimum: "):
    pool.addETHLiquidityWithProof(
        [],
            # Empty proof
        0, 0, 0, 0, # Dummy position values
        {"from": other, "value": 1e14}
    )

tx = pool.addETHLiquidityWithProof(
    [],
        # Empty proof
    0, 0, 0, 0, # Dummy position values
    {"from": other, "value": 1e15}
)

assert tx.events['ETHLiquidityDeposited'][0]['user'] == other
assert tx.events['ETHLiquidityDeposited'][0]['ethAmount'] == 1e15

def test_remove_liquidity(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)

    # Deploy uniswap
    factory = deploy_factory(owner, owner)

```

```
weth = deploy_weth(owner)

router = deploy_router(owner, factory.address, weth.address)

token1 = deploy_erc(owner, "token1", "token1")
factory.createPair(weth.address, token1.address)

maker_factory, pool, _ = deploy_pool(owner, token1.address, router.address)

# mint some tokens

token1.mint(other, 10e18)
token1.mint(extra, 10e18)
token1.approve(pool, 10e18, {"from": other})
token1.approve(pool, 10e18, {"from": extra})

pool.addTokenLiquidityWithProof(
    2e18,
    [],
        # Empty proof
    0, 0, 0, 0, # Dummy position values
    {"from": other}
)
pool.addTokenLiquidityWithProof(
    2e18,
    [],
        # Empty proof
    0, 0, 0, 0, # Dummy position values
    {"from": extra}
)

with reverts("InsufficientWithdrawal: "):
    pool.removeTokenLiquidityWithProof(
        1000,
        3e18,
        [],
        0, 0, 0, 0,
        {"from": other}
)
```

```
with reverts("InvalidSharesPercentage: "):
    pool.removeTokenLiquidityWithProof(
        0,
        3e18,
        [],
        0, 0, 0, 0,
        {"from": other}
    )

    tx = pool.removeTokenLiquidityWithProof(
        5000,
        1e18,
        [],
        0, 0, 0, 0,
        {"from": other}
    )

    assert tx.events['TokenLiquidityWithdrawn'][0]['user'] == other
    assert tx.events['TokenLiquidityWithdrawn'][0]['token'] == token1.address
    assert tx.events['TokenLiquidityWithdrawn'][0]['tokenAmount'] == 1e18

with reverts("NoLiquidity: "):
    pool.removeETHLiquidityWithProof(
        1000,
        3e18,
        [],
        0, 0, 0, 0,
        {"from": other}
    )

with reverts("InvalidSharesPercentage: "):
    pool.removeETHLiquidityWithProof(
        0,
        3e18,
        [],
        0, 0, 0, 0,
        {"from": other}
```

```

        )

pool.addETHLiquidityWithProof(
    [],           # Empty proof
    0, 0, 0, 0,   # Dummy position values
    {"from": other, "value": 1e15}
)

pool.addETHLiquidityWithProof(
    [],           # Empty proof
    0, 0, 0, 0,   # Dummy position values
    {"from": extra, "value": 4e15}
)

tx = pool.removeETHLiquidityWithProof(
    5000,
    0.5e15,
    [],
    0, 0, 0, 0,
    {"from": other}
)

assert tx.events['ETHLiquidityWithdrawn'][0]['user'] == other
assert tx.events['ETHLiquidityWithdrawn'][0]['token'] == token1.address
assert tx.events['ETHLiquidityWithdrawn'][0]['ethAmount'] == 0.5e15

def test_buy_sell_order(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)

    # Deploy uniswap
    factory = deploy_factory(owner, owner)
    weth = deploy_weth(owner)
    router = deploy_router(owner, factory.address, weth.address)

```

```

token1 = deploy_erc(owner, "token1", "token1")
factory.createPair(weth.address, token1.address)
maker_factory, pool, orderbook = deploy_pool(owner, token1.address,
router.address)

# mint some tokens
token1.mint(other, 10e18)
token1.mint(extra, 10e18)
token1.approve(pool, 10e18, {"from": other})
token1.approve(pool, 10e18, {"from": extra})

pool.addTokenLiquidityWithProof(
    2e18,
    [],
        # Empty proof
    0, 0, 0, 0, # Dummy position values
    {"from": extra}
)

pool.addETHLiquidityWithProof(
    [],
        # Empty proof
    0, 0, 0, 0, # Dummy position values
    {"from": other, "value": 1e15}
)

with reverts("OnlyOrderbook: "):
    pool.executeBuyOrder(1e16, 1e18, bytes(32), {"from": owner})
with reverts("AmountZero: "):
    pool.executeBuyOrder(0, 1e18, bytes(32), {"from": orderbook.address})
with reverts("AmountZero: "):
    pool.executeBuyOrder(1e16, 0, bytes(32), {"from": orderbook.address})
with reverts("ETHAmountMismatch: "):
    pool.executeBuyOrder(1e16, 1e18, bytes(32), {"from": orderbook.address})

owner.transfer(orderbook.address, "1 ether")
with reverts("InsufficientTokenLiquidity: "):

```

```

        pool.executeBuyOrder(1e16, 3e18, bytes(32), {"from": orderbook.address,
"value": 1e16})

        tx = pool.executeBuyOrder(1e15, 1e18, bytes(32), {"from": orderbook.address,
"value": 1e15})

        assert tx.events['PoolBalanceUpdated'][0]['token'] == token1.address
        assert tx.events['PoolBalanceUpdated'][0]['newTotalEth'] == 2e15
        assert tx.events['PoolBalanceUpdated'][0]['newTotalTokens'] == 1e18

    with reverts("OnlyOrderbook: "):
        pool.executeSellOrder(1e16, 1e18, bytes(32), {"from": owner})
    with reverts("AmountZero: "):
        pool.executeSellOrder(0, 1e18, bytes(32), {"from": orderbook.address})
    with reverts("AmountZero: "):
        pool.executeSellOrder(1e16, 0, bytes(32), {"from": orderbook.address})
    with reverts("InsufficientETHLiquidity: "):
        pool.executeSellOrder(3e15, 2e18, bytes(32), {"from": orderbook.address})

    token1.approve(pool, 0.5e18, {"from": orderbook.address})

    tx = pool.executeSellOrder(1e15, 0.5e18, bytes(32), {"from": orderbook.address})
    assert tx.events['PoolBalanceUpdated'][0]['token'] == token1.address
    assert tx.events['PoolBalanceUpdated'][0]['newTotalEth'] == 1e15
    assert tx.events['PoolBalanceUpdated'][0]['newTotalTokens'] == 1.5e18

def test_set_min_liquidity(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)

    # Deploy uniswap
    factory = deploy_factory(owner, owner)
    weth = deploy_weth(owner)
    router = deploy_router(owner, factory.address, weth.address)

```

```

token1 = deploy_erc(owner, "token1", "token1")

maker_factory, pool, _ = deploy_pool(owner, token1.address, router.address)

with reverts():
    pool.setMinLiquidity(1e16, {"from": other})

    with reverts("InvalidMinLiquidity: "):
        pool.setMinLiquidity(0, {"from": owner})

assert pool.minLiquidity() == 1e15
tx = pool.setMinLiquidity(1e16, {"from": owner})
assert tx.events['MinLiquidityUpdated'][0]['newMinLiquidity'] == 1e16
assert pool.minLiquidity() == 1e16

def test_set_min_token_liquidity(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)

    # Deploy uniswap
    factory = deploy_factory(owner, owner)
    weth = deploy_weth(owner)
    router = deploy_router(owner, factory.address, weth.address)

    token1 = deploy_erc(owner, "token1", "token1")
    maker_factory, pool, _ = deploy_pool(owner, token1.address, router.address)

    with reverts():
        pool.setMinTokenLiquidity(1e18, {"from": other})

        with reverts("InvalidMinTokenLiquidity: "):
            pool.setMinTokenLiquidity(0, {"from": owner})

    assert pool.minTokenLiquidity() == 2e18
    tx = pool.setMinTokenLiquidity(1e18, {"from": owner})

```

```
assert tx.events['MinTokenLiquidityUpdated'][0]['newMinTokenLiquidity'] == 1e18
assert pool.minTokenLiquidity() == 1e18

def deploy_pool(owner, token, router):
    # Deploy registry
    gradient_token = deploy_erc(owner, "gradient", "gradient")
    orderbook = deploy_orderbook(owner)
    registry = deploy_registry(owner)
    fallback_executor = deploy_fallback_executor(owner, registry.address)

    aggregator = deploy_event_aggregator(owner)
    maker_factory = deploy_maker_factory(owner, registry.address,
                                         aggregator.address)

    registry.setMainContracts(
        maker_factory.address,
        gradient_token.address,
        orderbook.address,
        fallback_executor.address,
        router,
        {"from": owner}
    )

    maker_factory.createPool(token, {"from": owner})
    pool_address = maker_factory.getPool(token)
    pool = GradientMarketMakerPoolV2.at(pool_address)
    return maker_factory, pool, orderbook
```

Technical Findings Summary

Findings

Vulnerability Level	Total	Pending	Not Apply	Acknowledged	Partially Fixed	Fixed
HIGH	1				1	
MEDIUM	5					5
LOW	5				1	4
INFORMATIONAL	1					1

Assessment Results

Score Results

Review	Score
Global Score	85/100
Assure KYC	Not completed
Audit Score	85/100

The Following Score System Has been Added to this page to help understand the value of the audit, the maximum score is 100, however to attain that value the project must pass and provide all the data needed for the assessment. Our Passing Score has been changed to 84 Points for a higher standard, if a project does not attain 85% is an automatic failure. Read our notes and final assessment below. The Global Score is a combination of the evaluations obtained between having or not having KYC and the type of contract audited together with its manual audit.

Audit PASS

Following our comprehensive security audit of the token contract for the Gradient project, the project did meet the necessary criteria required to pass the security audit.

Disclaimer

Assure Defi has conducted an independent security assessment to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the reviewed code for the scope of this assessment. This report does not constitute agreement, acceptance, or advocating for the Project, and users relying on this report should not consider this as having any merit for financial adGradient in any shape, form, or nature. The contracts audited do not account for any economic developments that the Project in question may pursue, and the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude, and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are entirely free of exploits, bugs, vulnerabilities or depreciation of technologies.

All information provided in this report does not constitute financial or investment adGradient, nor should it be used to signal that any person reading this report should invest their funds without sufficient individual due diligence, regardless of the findings presented. Information is provided ‘as is, and Assure Defi is under no covenant to audit completeness, accuracy, or solidity of the contracts. In no event will Assure Defi or its partners, employees, agents, or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions or actions with regards to the information provided in this audit report.

The assessment serGradients provided by Assure Defi are subject to dependencies and are under continuing development. You agree that your access or use, including but not limited to any serGradients, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies with high levels of technical risk and uncertainty. The assessment reports could include false positives, negatives, and unpredictable results. The serGradients may access, and depend upon, multiple layers of third parties.