**MPO election Unit testing**

| The PBI, the Task Description (from Sprint Log) with Unique Testing Number: | MPO election: Unit Testing - Test Logs  MPO\_ELEC\_UT\_01 |
| --- | --- |
| Team Member(s) Responsible: | Derrick Dischinger |
| Inputs: | Mpo01, an MPO object created with 3 seats, 380 ballots, 10 candidates and 4 parties  Index01 the index to adjust |
| Tests: | 1. AdjustRemainingVotes correctly updates remainingVotes with a valid index 2. index = -1 which is not within the valid range IOException should be thrown 3. index = 15 which is not within the valid range 4. ensure that the correct value is being subtracted from the index |
| Outputs: | Success |
| Passed or Failed | 1. PASSED 2. PASSED 3. PASSED 4. PASSED |
| Date | April 13th 2024 |

| The PBI, the Task Description (from Sprint Log) with Unique Testing Number: | MPO election: Unit Testing - Test Logs  MPO\_ELEC\_UT\_02 |
| --- | --- |
| Team Member(s) Responsible: | Derrick Dischinger |
| Inputs: | Mpo01, an MPO object  Expected1 the list to compare to  Copied1, mpo01.deepCopyVotes(expected1) |
| Tests: | 1. ensure both arrays lists are the same after the deepCopyVotes is called 2. ensure variables can be changed in original and not affect copy 3. situation where an empty list is passed in 4. adding to one empty list will not affect the other |
| Outputs: | Success |
| Passed or Failed | 1. PASSED 2. PASSED 3. PASSED 4. PASSED |
| Date | April 13th 2024 |

| The PBI, the Task Description (from Sprint Log) with Unique Testing Number: | MPO election: Unit Testing - Test Logs  MPO\_ELEC\_UT\_03 |
| --- | --- |
| Team Member(s) Responsible: | Derrick Dischinger |
| Inputs: | Mpo01, an MPO object  Expected1, the list to compare to, manually constructed  Result1, mpo01.initializeSeatAllocation() |
| Tests: | 1. ensure the resulting array has the correct default values in all fields 2. manually compare because assertEquals compares memory addresses for theint[]s |
| Outputs: | Success |
| Passed or Failed | 1. PASSED 2. PASSED |
| Date | April 13th 2024 |

| The PBI, the Task Description (from Sprint Log) with Unique Testing Number: | MPO election: Unit Testing - Test Logs  MPO\_ELEC\_UT\_04 |
| --- | --- |
| Team Member(s) Responsible: | Derrick Dischinger |
| Inputs: | Mpo01, an MPO object  randomValue, created with mpo01.generateRandom(); |
| Tests: | 1. Check if generated random number is within the specified range 2. Check if multiple calls to the method generate different values 3. test if the random numbers are actually distributed evenly over a large set and not pseudo random testing for 90% accuracy |
| Outputs: | Success |
| Passed or Failed | 1. PASSED 2. PASSED 3. PASSED |
| Date | April 13th 2024 |

| The PBI, the Task Description (from Sprint Log) with Unique Testing Number: | MPO election: Unit Testing - Test Logs  MPO\_ELEC\_UT\_05 |
| --- | --- |
| Team Member(s) Responsible: | Derrick Dischinger |
| Inputs: | Mpo01, an MPO object  numTie, number of ties to break  Result, mpo01.breakTie |
| Tests: | 1. numTie is 3, a valid number of ties 2. numTie is 0, not a valid tiebreak 3. numTie is 15, beyond range of number of parties 4. numTie is 1, should always return 0 |
| Outputs: | Success |
| Passed or Failed | 1. PASSED 2. PASSED 3. PASSED 4. PASSED |
| Date | April 13th 2024 |

| The PBI, the Task Description (from Sprint Log) with Unique Testing Number: | MPO election: Unit Testing - Test Logs  MPO\_ELEC\_UT\_06 |
| --- | --- |
| Team Member(s) Responsible: | Derrick Dischinger |
| Inputs: | Mpo01, an MPO object  mpo01.winOrder |
| Tests: | 1. tests to see whether the list is empty on start and if calling it with a winner adds the correct winner 2. ensure an index that is too large will throw an exception 3. ensure an index that is <0 will throw an exception |
| Outputs: | Success |
| Passed or Failed | 1. PASSED 2. PASSED 3. PASSED |
| Date | April 13th 2024 |

| The PBI, the Task Description (from Sprint Log) with Unique Testing Number: | MPO election: Unit Testing - Test Logs  MPO\_ELEC\_UT\_07 |
| --- | --- |
| Team Member(s) Responsible: | Derrick Dischinger |
| Inputs: | Mpo01, an MPO object  firstRound, true boolean  mpo01.adjustSeatAllocation(index01, firstRound);  index |
| Tests: | 1. test whether adding a seat at a specified index updates value to 1 from empty seat allocation array 2. test that adding in a second round to the same object works 3. testing with an index under accepted range IOException should be thrown 4. testing with an index over accepted range IOException should be thrown |
| Outputs: | Success |
| Passed or Failed | 1. PASSED 2. PASSED 3. PASSED 4. PASSED |
| Date | April 13th 2024 |

| The PBI, the Task Description (from Sprint Log) with Unique Testing Number: | MPO election: Unit Testing - Test Logs  MPO\_ELEC\_UT\_08 |
| --- | --- |
| Team Member(s) Responsible: | Derrick Dischinger |
| Inputs: | Mpo01, an MPO object  firstRound, true boolean  mpo01.adjustSeatAllocation(index01, firstRound);  index |
| Tests: | 1. seat should be given to top voted candidate 2. Correctly allocates seats to the top 2 candidates 3. Correctly allocates seats to the top 2 candidates 4. second seat gets correctly allocated to party that has members available even if they didn't win votes 5. there is a tie in seat allocation |
| Outputs: | Success |
| Passed or Failed | 1. PASSED 2. PASSED 3. PASSED 4. PASSED 5. PASSED |
| Date | April 13th 2024 |

| The PBI, the Task Description (from Sprint Log) with Unique Testing Number: | MPO election: Unit Testing - Test Logs  MPO\_ELEC\_UT\_09 |
| --- | --- |
| Team Member(s) Responsible: | Derrick Dischinger |
| Inputs: | Mpo1, a new MPO election created using testFile02  Result, a ResultData object from mpo1.runElection() |
| Tests: | 1. mpo election with one seat to allocate, no ties, 2 parties. 5 candidates 2. normal mpo election with 3 seats to allocate, no ties, 4 parties, 10 candidates |
| Outputs: | Success |
| Passed or Failed | 1. PASSED 2. PASSED |
| Date | April 13th 2024 |