# Test Description

**Test Name or ID**: test\_addRoute\_valid

**Test Type**: Black box

**Description**: Validates adding a route to the map. Ensures that the route is correctly added to the map structure.

**Setup:** Initialize the map using populateMap(). Retrieve a valid route using getBlueRoute().

**Test Function**: test\_addRoute\_valid

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Adding a valid route to the map | Route from getBlueRoute() | Map updated with new route | Map updated correctly | Pass |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Bugs Found**:

**No bugs were found** during the execution of the test cases. All tests passed successfully.

# Test Description

**Test Name or ID**: test\_addRoute\_invalid

**Test Type**: Black box

**Description**: Validates handling of an invalid route to ensure the map remains unchanged

**Setup:** Initialize the map using populateMap(). Retrieve a route using getBlueRoute() and modify it to be invalid.

**Test Function**: test\_addRoute\_invalid

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Adding an invalid route to the map | Route with invalid starting point (-1, x) | Map remains unchanged | Map remained unchanged | Pass |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Bugs Found**: **No bugs were found** during the execution of the test cases. All tests passed successfully.

# Test Description

**Test Name or ID**: test\_addRoute\_overlap

**Test Type**: Black box

**Description**: Validates handling of overlapping routes to ensure correct map updating.

**Setup:** Initialize the map using populateMap(). Add two overlapping routes sequentially using getBlueRoute() and getGreenRoute().

**Test Function**: test\_addRoute\_overlap

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Adding overlapping routes to the map | Two overlapping routes | Map correctly updated with overlapping routes | Map updated correctly | Pass |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Bugs Found**: **No bugs were found** during the execution of the test cases. All tests passed successfully.

# Test Description

**Test Name or ID**: test\_populateMap

**Test Type**: Black box

**Description**: Validates map population to ensure correct dimension

**Setup:** populateMap()

**Test Function**: test\_populateMap.

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Populate map | None | Map with correct dimensions | Map has correct dimensions | Pass |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Bugs Found**:

**No bugs were found** during the execution of the test cases. All tests passed successfully.

# Test Description

**Test Name or ID**: test\_distance

**Test Type**: Black box

**Description**: Validates distance calculation between points.

**Setup:** Create two points with known coordinates.

**Test Function**: test\_distance

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Calculate distance between points | Points (0,0) and (3,4) | |  | | --- | | Distance is 5.0 |  |  | | --- | |  | | Distance calculated correctly | Pass |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Bugs Found**:

**No bugs were found** during the execution of the test cases. All tests passed successfully.

# Test Description

**Test Name or ID**: test\_shortestPath

**Test Type**: Black box

**Description**: Validates finding the shortest path between two points on the map.

**Setup:** Initialize the map using populateMap(). Define start and destination points.

**Test Function**: test\_shortestPath

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Find shortest path between points | Start (0,0) and destination (4,4) | Valid route found | Valid route found | Pass |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Bugs Found**: **No bugs were found** during the execution of the test cases. All tests passed successfully.

# Test Description

**Test Name or ID**: test\_getPossibleMoves

**Test Type**: Black box

**Description**: Validates getting possible moves from a given point.

**Setup:** Initialize the map using populateMap(). Define a point and a backpath point.

**Test Function**: test\_getPossibleMoves

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Get possible moves from a point | Point (1,1), backpath (0,0) | List of possible moves | Possible moves correctly identified | Pass |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Bugs Found**: **No bugs were found** during the execution of the test cases. All tests passed successfully.

# Test Description

**Test Name or ID**: test\_getClosestPoint

**Test Type**: Black box

**Description**: Validates finding the closest point in a route.

**Setup:** Define a route and a point to check.

**Test Function**: test\_getClosestPoint

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Find closest point in a route | Route with points, point (2,3) | Index of closest point | Closest point correctly identified | Pass |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Bugs Found**: **No bugs were found** during the execution of the test cases. All tests passed successfully.

# Test Description

**Test Name or ID**: TestEqPt\_EqualPoints

**Test Type**: Black box

**Description**: Validates equality check for equal points.

**Setup:** Define two equal points

**Test Function**: TestEqPt\_EqualPoints

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Check equal points | Points (1,2) and (1,2) | |  | | --- | |  |  |  | | --- | | Points are equal | | |  | | --- | |  |  |  | | --- | | Points are equal | | Pass |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Bugs Found**: **No bugs were found** during the execution of the test cases. All tests passed successfully.

# Test Description

**Test Name or ID**: TestEqPt\_NotEqualPoints

**Test Type**: Black box

**Description**: Validates equality check for non-equal points.

**Setup:** Define two different points.

**Test Function**: TestEqPt\_NotEqualPoints

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Check non-equal points | Points (1,2) and (1,5) | |  | | --- | | Points are not equal |  |  | | --- | |  | | Points are not equal | Pass |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Bugs Found**: **No bugs were found** during the execution of the test cases. All tests passed successfully.

# Test Description

**Test Name or ID**: TestEqPt\_NegativeCoordinates

**Test Type**: Black box

**Description**: Validates equality check with negative coordinates.

**Setup:** Define two points with negative coordinates..

**Test Function**: TestEqPt\_NegativeCoordinates

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Check points with negative coordinates | Points (1,2) and (-1,-2) | Points are not equal | Points are not equal | pass |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Bugs Found**:

**No bugs were found** during the execution of the test cases. All tests passed successfully.

### Summary

* **Total Test Cases**: 11
* **Passed**: 11
* **Failed**: 0