|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Data** | **Expected Result** | **Description** |
| Test 3 – Get Yellow route | struct Map baseMap = populateMap();  struct Route yellowRoute = getYellowRoute();  struct Map routeMap = addRoute(&baseMap, &yellowRoute);  bool routeAdded = false;  // Check if the route has been added correctly by verifying if any squares have the route symbol  for (int r = 0; r < routeMap.numRows; r++)  {  for (int c = 0; c < routeMap.numCols; c++)  {  if (routeMap.squares[r][c] == YELLOW)  {  routeAdded = true;  break;  }  }  if (routeAdded)  break;  }  //Assert  Assert::IsTrue(routeAdded); | Returns True | Uses the map and matches the points from the yellow function and returns true if all points are found |
| Test 4 – Get blue route | struct Map baseMap = populateMap();  struct Route blueRoute = getBlueRoute();  struct Map routeMap = addRoute(&baseMap, &blueRoute);  // Check if the route has been added correctly by verifying if any squares have the route symbol  bool routeAdded = false;  for (int r = 0; r < routeMap.numRows; r++)  {  for (int c = 0; c < routeMap.numCols; c++)  {  if (routeMap.squares[r][c] == BLUE)  {  routeAdded = true;  break;  }  }  if (routeAdded)  break;  }  //Assert  Assert::IsTrue(routeAdded); | Returns True | Search the map for all the implemented blue route, once all of the points are found the test returns true. |