London Pollution Viewer

22nd March 2025

Group Name: *The Fellowship of the Pollutants*

Names: Muhammad-Ali Mohsin, Parsa Pilever, Amirreza Ahadi Moghaddam, Mohammed Al Yahya

Student Numbers: 23147793, 24038447, 24000518, 23104701

Module Code: 4CCS1PPA

Assignment 4: London Air Pollution

King's College London

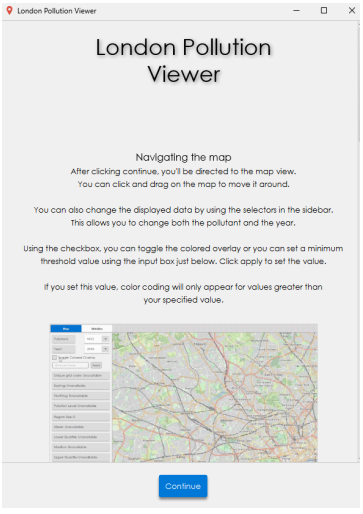
Group Member Contributions

A short description of which group members implemented which classes:

- Parsa Pilever - Implemented the LondonPollutionViewer, LondonPollutionView and Visualisation pane classes which involve all the outer GUI elements including the window, switching between windows and the tab-based view. Also responsible for all the CSS and styling and helped create the Sidebar class which contains all of the controls for interacting with the map.
- Mohammed Al Yahya - Implemented the InstructionPane class, which creates the instruction screen by which the user is first greeted. This provides detailed instructions on how to use the application and allows the user to then access the main application.
- Muhammad-Ali Mohsin - Implemented the MapPane and MapOverlay classes which involve showing the map image, allowing the user to view specific pollution metrics, colour coding to show different pollution levels and selecting custom regions to view region-specific statistics. Also implemented the DataManager class responsible for loading datasets, providing methods for accessing the data and calculating statistics about the data. Additionally, helped create the Sidebar class which contains all of the controls for interacting with the map and displaying pollution data to the user. Responsible for the DataLoaderTest class.
- Amirreza Ahadi Moghaddam - Implemented the PollutionStatsPane and AdditionalStatsPane classes. This is responsible for showing different graphs and statistics about the pollution data such as the mean values of pollutants over time or the minimum and maximum pollution values for different years. Responsible for the DataManagerTest class.

Description of the GUI and its functionality

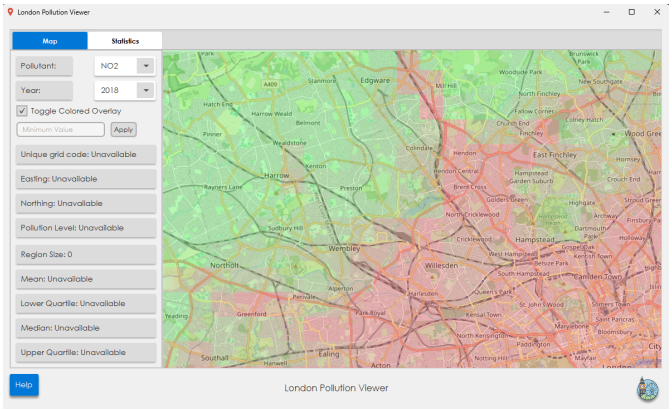
There are a total of three screens. You are first directed to the instruction screen which then leads to the main London Pollution Viewer screen. From here, there are tabs allowing you to switch between the map and statistics panels:



- Instruction Screen - The user is directed to the instruction screen when they start up the application. This screen provides a comprehensive guide to using the program with the aid of gifs that display a demonstration of the program. The user can scroll through all the instructions and once they are ready, they can press the continue button to progress to the main application. They can return to this screen at any point by clicking the help button from within the main application.

- London Pollution View - The London Pollution View is the main view of the program. It consists of a tab layout with two tabs, a Map tab and a Statistics tab. There is also a help button at the bottom of the screen along with a label and an icon. This is the outer GUI layout and is visible even when switching between tabs.

- Map Tab - The Map Tab is the default tab when the user accesses the main application. It contains the following:
    - A map image which the user can pan around by clicking and dragging. Hovering over an area on the map also shows information about that area.
    - A coloured overlay on the map which tints areas with higher pollution in red and areas with lower pollution in green.
    - Dropdown menus, allowing you to switch the pollutant type and the year of the currently viewed dataset.
    - A checkbox which allows the colored overlay to be toggled on and off.
    - A text box that allows you to enter a minimum threshold value for the pollution. If you enter a number and click the apply button, the colored overlay will apply a filter and only show areas with a pollution value higher than your inputted data point.
    - Labels for a marker. You can double-click on the map to place down a marker and view the data for that location in the sidebar.
    - Region selection labels. You can hold down shift and drag your cursor over the map to select a region. The labels will update as you select your region providing you with statistics such as the mean, median and quartiles. To deselect an area, hold control (command on Mac) and drag your cursor over the area.



- Statistics Tab - The Statistics Tab can be accessed using the tab menu at the top of the main screen. It contains the following:
    - A central graph showing the currently selected data type. It has pollution values on the y-axis and years on the x-axis allowing it to show trends over time.
    - A drop-down menu on the left, allowing the user to switch between three different types of graphs. They can view the average pollution values throughout the years, the maximum pollution values or the minimum pollution values.

- Checkboxes below the graph which allow the user to filter which pollutants are shown on the graph. They can select multiple pollutants at the same time and have multiple lines shown on the same graph.
- An additional stats button which opens up another window. This window has a drop-down menu that allows the user to select a year and it shows additional statistics for that year. This includes the maximum and minimum pollution values for each pollutant as well as the location at which they occurred.

Challenge Tasks:

- Custom filtering for pollution values which allows the user to specify a threshold and have only values greater than the threshold displayed on the map. This would allow the user to highlight specific areas and identify pollution hotspots.
- Area selection which allows the user to draw a custom region on the map and view statistics for that specific region. This includes the mean, median and more. This will allow the user to have detailed stats in order to make a better decision based on these data.
- More extensive graph-based trends of pollution over time. (Minimum values, maximum values, and average values). The user is able to select among these graphs and also to filter the data shown on the graph based on the pollutants.
- Comparison between different years of the pollution data represented on the graph. This would allow the user to have more thorough data about the locations presented on the map. This will allow the user to make a better use of the application.

Unit Testing

We conducted lots of testing throughout the application and wrote test classes for two of our classes which different implement any GUI functionality:

- The DataManager class is responsible for loading datasets when provided with a pollutant type and year. It then caches these datasets and provides methods to access them and calculate different statistics about them. We wrote some tests to ensure these statistics were correct and data was correctly retrieved:
    - testLoadDataSet - This test involved passing in a valid pollutant and year and ensuring that the dataset was successfully loaded without errors.
    - testLoadNonExistentDataSet - This test involved passing in invalid parameters and ensuring that null was returned by the data manager signifying that no dataset could be loaded.
    - testGetDataPoint - This test involved picking a valid point on the map and ensuring that the data point at that location could be successfully retrieved.
    - testGetDataPointNotFound - This test involved picking an invalid data point on the map and ensuring that null was returned signifying the data point could not be found.
    - testFindMaxPollution - This test involved ensuring the findMaxPollution method returned the correct maximum pollution value in the dataset.
    - testFindMinPollution - This test involved ensuring the findMinPollution method returned the correct minimum pollution value in the dataset.
    - testGetMean - This test involved ensuring that the correct mean value was calculated by the getMean method.

- The DataLoader class is responsible for loading in the raw data. It is used by the DataManager and is mainly responsible for parsing the csv data into actual data sets. It also makes sure the data is within the bounds of the map. We wrote the following tests:
    - testLoadDataFileValid - This test involved loading in a data set and ensuring that all of the data such as the units, pollutant and year were correctly labelled.
    - testLoadDataPointsInBounds - This test involved loading in a dataset and making sure the points in the dataset were all within the bounds of the map. This reduces the size of the data set allowing for more efficient searching and statistics calculations.

We had many failed tests particularly in the DataManager test class where we did not get the desired outcomes but by working through them, we were able to fix any bugs and pass all tests successfully.