

# 5011CEM BIG DATA PROGRAMMING PROJECT

---



## EUROPEAN OZONE VISUALISATION PORTFOLIO

**Submission Date: 24<sup>th</sup> April 2020**

## TABLE OF CONTENTS

Project Management .....	4
Time Management.....	5
Version control .....	6
Automated Testing .....	6
List of References.....	7
List of Appendices.....	8
Appendix A.....	8
Appendix B.....	10
Appendix C.....	11
Appendix D.....	12
Appendix E .....	13
Appendix F .....	14
Appendix G.....	15
Appendix H.....	16
Sprint Planning.....	16
Sprint Scum Meetings .....	17
Sprint Review .....	18
Sprint Retrospective .....	18
Appendix I .....	19
Sprint Planning.....	19
Sprint Scrum Meetings.....	20
Sprint Review .....	21
Sprint Retrospective .....	22
Appendix J.....	22

Appendix K .....23

## PROJECT MANAGEMENT

The project management involved planning the project, controlling the development and updating the project plan, if changes were needed. Firstly, an initial specification was drawn up, which was presented to the client for confirmation, to ensure the project's aim and specifications were in line with the client's wishes. After the specifications were agreed upon by both sides, the specification was updated to match the agreed requirements.

Afterwards, the Agile methodology was chosen to be the method of iterative planning and control of the project. This meant, each SMART target from the specification was placed within a Product Backlog (Appendix G) and prioritized, the top of the stack being the highest priority tasks. This allowed for clear and a concise understanding of what tasks needed to be completed to fully develop this project.

For a more in-depth explanation of how the Agile methodology was implemented, refer to the Time Management section of this document. However, when each task from the Product Backlog was taken for the weekly sprint, a flowchart of the feature was first created (Appendix B, C, D, E, F). These, flowcharts allowed for a clear visual representation of how each feature's processes should work, without the need for any confusing syntax and technical detail. Thus, leading to significantly simpler and more efficient code development, due to the process flow already being defined.

To document the current progress of the project, a Product Log was used (Appendix A). It recorded all completed tasks, with explanation of the work achieved and any challenges or specific notes, as well as the current task being implemented. This clearly displayed the project's current progress, enabling any individual to read the document, and to quickly understand what tasks have been completed, which task was being developed, and which are to still be completed. This is essential in a professional environment, due to the volatility of developers leaving or becoming unable to carry on with development therefore, requiring a new developer needing to quickly understand the current project's progress, to be able to continue developing the project.

The Project Log (Appendix A) was populated every time development on the project was undertaken. At the start of each Agile sprint, the start date and task were written down. Then during the implementation of the task, notes were written down, alongside any challenges faced. These notes could then be later used to address other tasks and challenges faced. Finally, once tasks were completed, the work implemented would be written down, with its completion date, alongside the next task to be completed.

Any alteration to the project plan would also be written down within the Project Log (Appendix A). An example of this was the '12/03/2020' task (Appendix A), this task was initially undertaken, but due to time constraints and complications, it was agreed to be dropped from development. This was noted and the project plan was altered, being to move onto the next task. Another example of this was the '03/03/2020' task (Appendix A), this task was initially not planned within the given features sprint however, after testing, the feature did not meet the 0.5 seconds execution target agreed upon in the specification. Therefore, the project plan was altered, delaying the next task, in order to go back and alter the feature to meet the 0.5 seconds target.

This documentation of the Agile methodology tasks, flowcharts and Project Log, would then be repeated for the next task, until all tasks were completed.

## TIME MANAGEMENT

The Agile methodology was adopted to structure the management of the project, including time management. Agile was chosen due to its emphasis on frequent release of features, allowing for rapid adaptability of the project at any stage, unlike methodologies such as Waterfall model, which is very difficult to make changes once the code development has started (Holcombe 2008:1-18). It also grants the client the ability to work alongside the developers, allowing the client the opportunity to change their requirements and steer of the project to their wishes (Holcombe 2008:1-18). However, due to this project being independent, the project manager, scrum master and team members was myself.

The Agile methodology consists of a Product Backlog, Sprint Backlog, a sprint, Scrum meeting and retrospective meetings (Holcombe 2008:1-18). As mentioned in the Project Management section, the Smart targets from the specification were placed within a Product Backlog (Appendix G) and prioritized, the top of the stack being the highest priority tasks. This was to initially start the Agile process and afterwards, it was agreed to have weekly sprints of 7 days, due to the time constraint of the 10-week deadline. Additionally, due to other commitments such as, other module coursework and exams, not every day will be used to develop the project, instead 2 to 3 days a week were designated to this project, which varied each week depending on the given week's priorities.

At the start of each Sprint, a Sprint Planning meeting was held (Appendix H and I) to discuss how the task from the Product Backlog was to be broken down into smaller tasks and placed with a Sprint Backlog, in priority order. A task from the Sprint Backlog was taken each day and a scrum meeting was held before any work was started (Appendix H and I), this was to discuss what work was carried out during the previous workday, and any current standing issues regarding the project was noted down in the meeting brief. This was to then be used to re-order sprint backlog task priorities, if needed. Then the top task on the sprint backlog was taken off, to then be developed for the current day. After each day of development, any progress and issues were documented in the Project Log (Appendix A), to be used in the next scrum.

Once the weekly sprint was over, a review meeting (Appendix H and I) and retrospective meeting (Appendix H and I) was held to discuss the sprints progress and to address any issues that occurred or were still standing. This would then be used to adjust the product backlog if needed, to either add tasks to fix issues, add tasks upon the client's requests, or to re-priorities the tasks.

This Agile methodology is iterative therefore, this process was repeated for each task in the product backlog, until all the tasks were completed, with the final product finalized and ready for the client.

## **VERSION CONTROL**

Version control is essential for project development, it reduces the complications of errors and incorrect implementation of features, due to its ability to roll-back code to the last previous working version. For this project, GitHub was the method of version control. To start, a new repository was created for this project and the first initial attempt at code development was uploaded to the master branch. Afterwards, when each feature/task was being developed, a branch would be created, designated to the given feature (Appendix K). This branch would act as the buffer between the working master branch, and the volatile working branch. After each section of the feature was developed, it would be pushed to its feature branch. Only once the feature was fully developed and tested, could the branch be merged with the master branch, via a Pull Request, and then the branch would be deleted. This meant the current working version of the project would be updated, now including the new working feature. This process would be repeated for each development of a feature (Appendix J).

## **AUTOMATED TESTING**

This product requires 3 inputs. These are; user's choice for colourblind mode, user's choice for movie mode and the CBE ozone data files. If these inputs were invalid, then it would result in the program crashing. To avoid this issue, the code regarding input, has been error trapped. The first two inputs, being the user's choice for colourblind mode and movie mode have been designed in a way to prevent invalid input. This is due to the question only allowing for a pre-defined answer to be inputted, 'Yes' or 'No'. Therefore, invalid input is not possible, thus preventing possible program failure. As for the CBE ozone files, if an error occurs during importing a CBE file, a warning is displayed to the terminal explaining the error and the program exists. Therefore, preventing the program from crashing. Also, when plotting the data, if an error occurs, the error is caught, and a warning is sent to the terminal explaining the error and the program exits.

## **LIST OF REFERENCES**

Holcombe, M. (2008) *What is an Agile Methodology?* Hoboken, NJ, USA: John Wiley & Sons, Inc., 1-18

# LIST OF APPENDICES

## APPENDIX A

### Project Log

Date Planned	Expected Start Date	Date Achieved	Work Planned	Work Achieved	Next Steps	Notes, challenges, special info
20/01/2020	20/01/2020	20/01/2020	Module intro lecture	Listened to the entire lecture	Decide on a project choice by next week and play around with the data	
22/01/2020	22/01/2020	23/01/2020	Download Data Open the data files in MATLAB and manipulate the data	Data files downloaded Used in-built functions to change the data representation	Try and manipulate the data Choose a project	All in-built functions are on the MATLAB website
25/01/2020	25/01/2020	26/01/2020	Choose a project	Chose data visualisation to be my project	Go to the next lecture	
27/01/2020	27/01/2020	27/01/2020	Specification lecture	Learnt how to start writing my specification	Start writing my specification	
	29/01/2020	30/01/2020	Start writing my specification	SMART list created	Fill out rest of spec document template	Not all project requirements are easy to define in a measurable way
	03/02/2020	06/02/2020	Fill out rest of spec document template	Notes completed	Format the document to a higher standard later on	Overall info complete however, the content will need to be re-formatted to a professional standard later on
02/02/2020	02/02/2020	02/02/2020	Code planning lecture	Learnt how to structure the planning of my	Start planning my project	
20/02/2020	20/02/2020	23/02/2020	Go through the installation setup, write each step down	Set up MATLAB with the needed packages and converted the fig files to csv, writing down each step	Modify the CBE example code to import the csv files and use that data	The file to convert the .fig to .csv needed to change the \ to / as using Linux, not Windows
23/02/2020	23/02/2020	24/02/2020	Import the csv files to the example code to use the correct ozone data	Set up the code to import a given CBE csv file and then it displays these ozone values on the map of Europe	Modify the code to use a button to step-through each CBE hour file, so that the map shows each hour's ozone level in chronological order	Took a while to understand how the meshes worked and how to import and use the Z axis data which is used to display the ozone levels
27/02/2020	27/02/2020	02/03/2020	Create a button that steps through each hour csv file, which loads each file's ozone data and updates the map live, to display the new ozone figures	Each ozone hour data loads in sequence, being a right arrow key loads the next hour and a left arrow key loads the previous hour and the map updates.	Create a function to zoom in and out of the map to gain a closer look at the a local area's ozone data.	Struggled loading each .csv file one by one as I could not find any code online. However, I remembered the code to convert the fig files to .csv had a file loader already made, so I re-used that code by stripping it down to use the read in .csv files only. Also, finding code to get keypresses took longer than expected as many versions of code did not work.
03/03/2020	03/03/2020	04/03/2020	Planned in-between tasks as I wanted to modify previous task. Modify the code so only the ozone data is changed when each hour is loaded, leaving all other data, the Europe map etc all static. Allowing for a more efficient program.	The ozone data is the only data changed when the arrow keys load the next or previous hour. The map of Europe etc are only loaded at initial start-up and then remain loaded.	Create a function to zoom into a city node to gain a closer look at the local area's ozone data.	At first, the ozone hour data was just being displayed on top of each other, and the old data was not being cleared, leaving a map with over 25 hours of data all layered on top of each other. After talking to Mark, my lecturer, he suggested to use graphic object handles and set the ozone data specifically, leaving the other data loaded.
07/03/2020	07/03/2020	10/03/2020	Allow the user to zoom in and out of the map, to be able to get a closer view on certain countries, cities etc. down to each data pixel.	The map can be zoomed in and out of, down to each data plot pixel, however the map is plot is just scaled up to zoom in, and not replotted with the lat and lon being adjusted.	Try and stop the map from just being scaled up to zoom in, instead replot the map within the smaller range to allow for actual zoom, with the lat and lon being adjusted.	Allowing for basic zoom in and out feature was very simple.



12/03/2020	12/03/2020	Dropped	Allow the actual map to be zoomable, with lat and lon being adjusted on each zoom.	Managed to find geoaxes which has this feature built-in. However, I could not find how to plot the ozone data onto the new map.	Carry on with the remaining tasks, abandoning this task.	I searched for a very long time to try and find a feature which allowed for zoom, adjusting the lat and lon of the map live, according to what area the current map is plotting. I found geoaxes which did this, but i could not manage to merge my old code with this. I then tried to see if i could just make the feature myself, however I could not find any tutorials, forums on how to do this, therefore, after days of searching, I decided to abandon this task as it was taking up too much of my time, with no progress to show.
15/03/2020	23/03/2020	24/03/2020	Allow the user to choose a colourblind mode if they were colourblind. This would change the colourmap to a colour scheme that would be more suitable.	The program asks the user if they want colourblind mode upon loading up the program. If the user chooses yes, then a colourmap suitable for all colourblindness would be loaded, white to dark grey. If the user chose no, then the standard colourmap would be loaded.	Movie mode - the map does not need keyboard input to load each ozone hour, it automatically loads them incrementally in a loop.	No trouble with creating a popup question and changing the colourmap. The only time-consuming task was finding a colourmap scheme that would be suitable for the majority of colourblindness. After researching, I thought using a white to dark grey colour scheme would be the best because it would be suitable for all colourblindness as there is no colour involved.
26/03/2020	26/03/2020	26/03/2020	Movie Mode - automatically loading each ozone hour file incrementally in a loop.	The user is given a popup question at the start of the program for them to choose between standard step mode (original), or the new movie mode. It then loads the given choice.	Convert the one script into functions, to keep with coding standards.	I knew how to create a popup question, as it was in my last task. The movie mode feature was much easier than the original mode, because it just loads each ozone hour file incrementally in a loop.
27/03/2020	27/03/2020	27/03/2020	Convert the one script into functions, to keep with coding standards.	Created main(), Setting(), movieMode() and stepMode() functions to correctly split up each process.	Fully comment and implement docstrings to the code.	Initially, trying to understand and implement MatLab functions was a bit difficult. The reason was because my file was just one big script and so when I tried to create some functions within it, it just returned errors, which I had no idea how to fix. After some research, I realised that it was because the structure of my script was very messy and had incorrect endings to if statements etc. and was very hard to unpick and restructure. Therefore, I just made a new file and slowing converted each part into functions in the new file. One-by-one, I created each function and by the end, the program worked exactly the same as before but now using functions instead of just one big script.
30/03/2020	30/03/2020	30/03/2020	Fully comment and implement docstrings to the code.	All functions now have docstrings, stating what the function does, it's inputs and references to any sources I used. Each line has a comment, explaining what it does, when an explanation is needed.	Formalise the spec document to a high standard	No issues.
01/04/2020	24/04/2020		Formalise spec document text to a high standard	Spec document re-worded to a professional standard	Write project report	N/A
06/04/2020	06/04/2020	10/04/2020	Write project report	Project report fully written to a high standard	Write project portfolio	Took a very long time to write up, as so much explanation is needed to be documented
11/06/2020	11/06/2020		Write project portfolio	Project portfolio completed	Viva video	All evidence of the project planning has been put into the portfolio
21/04/2020	21/04/2020	21/04/2020	Make the Viva video	Viva video completed and submitted	Project finished	No issues.

Figure 1 – Project Log

## APPENDIX B

### Loading basemap and Ozone data

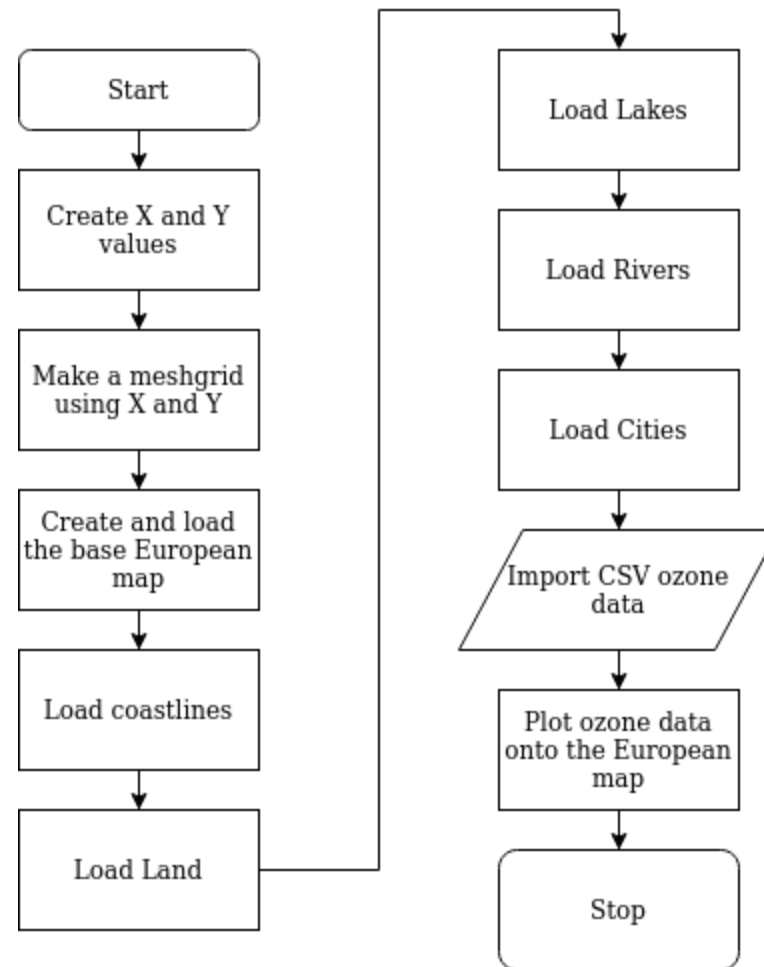


Figure 2 - The flowchart for the original overall script.

## Settings

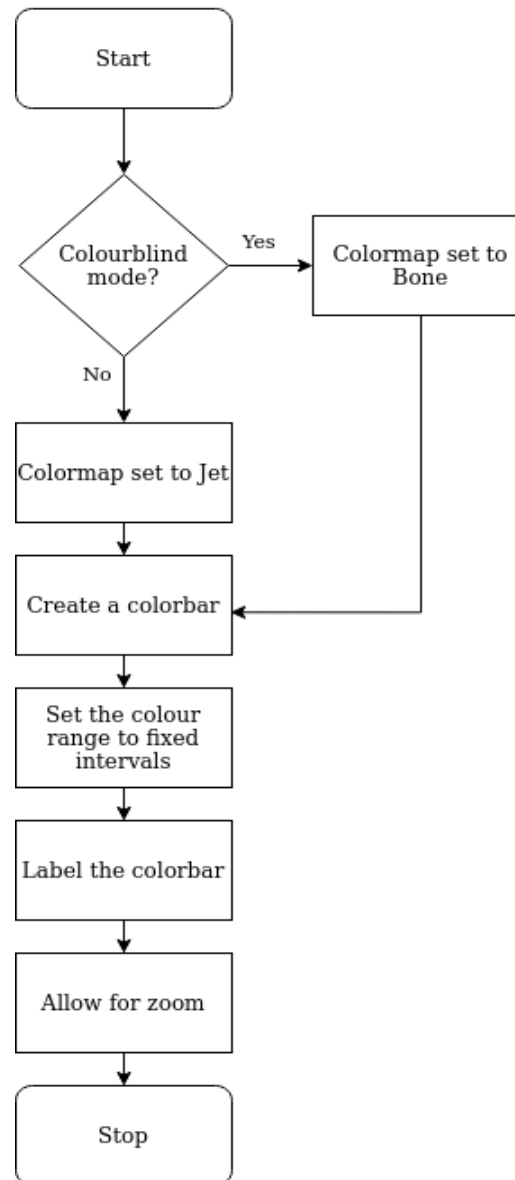


Figure 3 - The flowchart for the settings function.

# Loading Each ozone hour in step mode

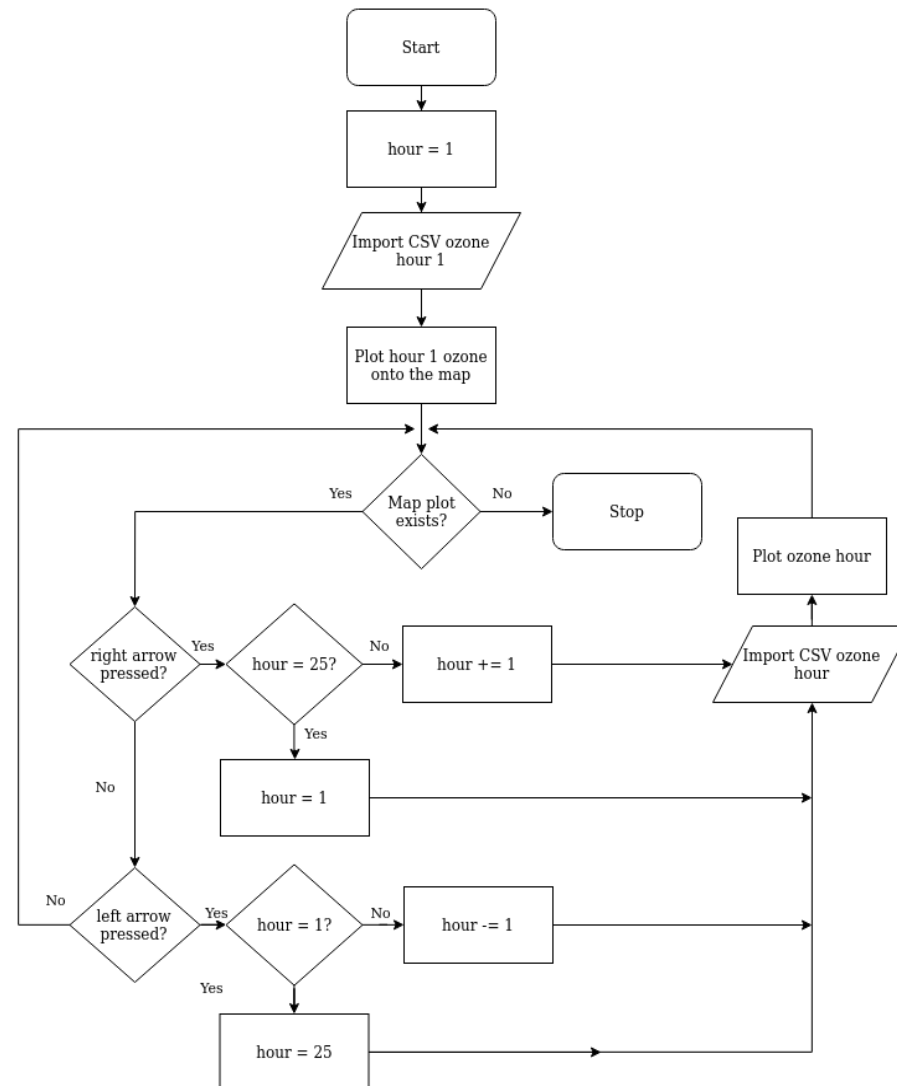


Figure 4 - The flowchart for the stepMode function.

## APPENDIX E

### Loading Each ozone hour in Movie mode

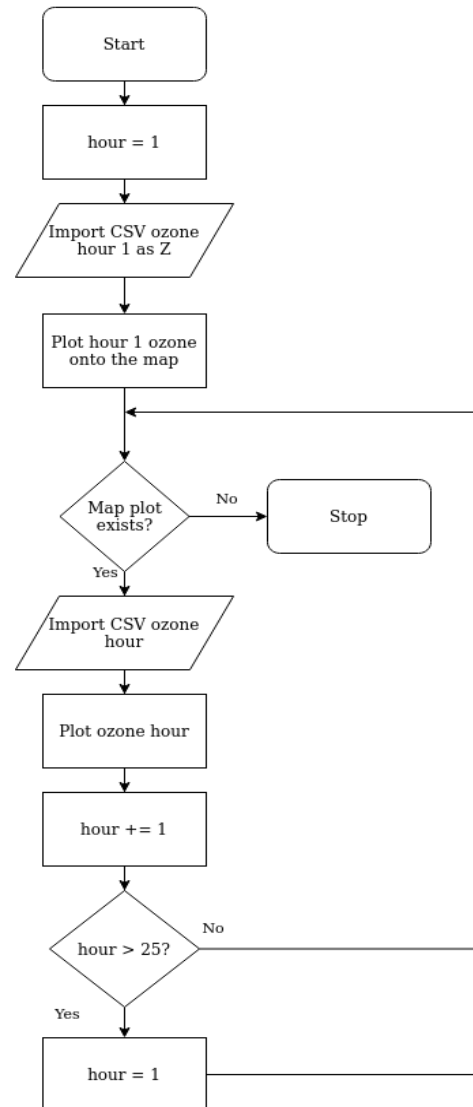


Figure 5 - The flowchart for the movieMode function.

## APPENDIX F

### Overall Project

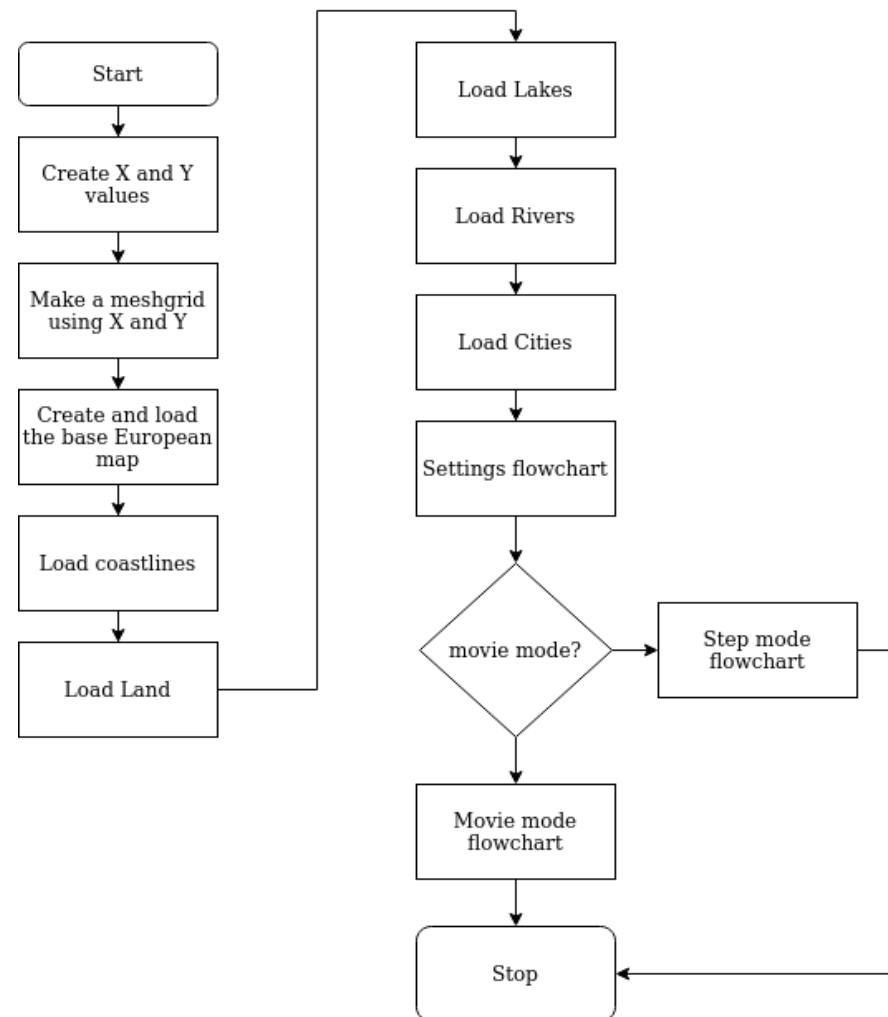


Figure 6 - The flowchart for the overall project, being the main function.

## APPENDIX G

Product Backlog
Create a button that steps through each hour csv file, which loads each file's ozone data and updates the map live, to display the new ozone figures
Modify the code so only the ozone data is changed when each hour is loaded, leaving all other data, the Europe map etc all static. Allowing for a more efficient program.
Allow the user to zoom in and out of the map, to be able to get a closer view on certain countries, cities etc. down to each data pixel.
Allow the actual map to be zoomable, with lat and lon being adjusted on each zoom.
Allow the user to choose a colourblind mode if they were colourblind. This would change the colourmap to a colour scheme that would be more suitable.
Movie Mode - automatically loading each ozone hour file incrementally in a loop.
Convert the one script into functions, to keep with coding standards.
Fully comment and implement docstrings to the code.
Figure 7 - The product Backlog of project.

## APPENDIX H

Evidence of Agile processes, sprint week 4.

### SPRINT PLANNING

Project Task	Sprint tasks	Estimated Time in Hours	Sprint Backlog	Estimated Time in Hours
Allow the actual map to be zoomable, with lat and lon being adjusted on each zoom.				
	Research feature implemeation	4	Research feature implemeation	4
	Map longitude and latitude update when zooming, to meet the new displayed map zone	4	Map zooms in wihtout boarder issue	4
	Map zooms in without boarder issue	4	Map longitude and latitude update when zooming, to meet the new displayed map zone	4

Figure 8 - The Sprint planning meeting for Sprint week 4, breaking down the Product Backlog feature into smaller tasks for the Sprint backlog



## SPRINT SCUM MEETINGS

Date	Sprint Week	Sprint Task	Scrum Meeting	Work Achieved Since Last Scrum	Notes, challenges, special info	Current Daily Task
12/03/2020	4	Research feature implementation	1	New Sprint	First Scrum meeting. Clear understanding of what task to be implemented today.	Research MatLab on how to address this feature
13/03/2020	4	Map zooms in without boarder issue	2	No feature found within Mapping Toolbox. However, Geoaxes offers the exact feature needed.	The issue is how to merge existing code to work for geoaxes.	Try and merge existing code to work alongside geoaxes
16/03/2020	4	Map zooms in without boarder issue	3	No progress made on merging the code to work alongside geoaxes	There is little documentation on how to use geoaxes with raw data. Therefore, trying to create a geodensityplot with the ozone data is proving to be a very big challenge, so far no progress has been made	Carry on trying to merge the existing code to work with geoaxes
18/03/2020	4	Map zooms in without boarder issue	4	Very little progress made. Created a geodensity plot on different data but not data of the same format as the ozone data	Decided to create a geodensityplot and then try to understand how it worked with the data. However, still no understanding on how to use it to address this problem.	Carry on trying to merge the existing code to work with geoaxes
20/03/2020	4	Map zooms in without boarder issue	5	Still no progress made	Simply there is not enough time or resources to understand how geoaxes works within this weekly sprint. An entire week's sprint has been used to try and implement this	One final attempt at making any progress as trying to get this feature to work

Figure 9 – Each SCRUM meeting from the sprint week 4

## SPRINT REVIEW

Date	Sprint Week	Sprint task	Tasks Completed	Product Backlog Updates	Next Sprint Plan Date
21/03/2020	4	Allow the actual map to be zoomable, with lat and lon being adjusted on each zoom.	No feature was completed	the Product Backlog remains the same. The task was not added back into the Project Backlog as it has been decided to drop the task.	23/03/2020

Figure – The Sprint Review meeting from Sprint week 4

## SPRINT RETROSPECTIVE

Date	Sprint Week	Sprint task	Self Analysis	Notes, challenges, special info	Improvements
21/03/2020	4	Allow the actual map to be zoomable, with lat and lon being adjusted on each zoom.	This sprint was a failure, due to the feature failing to be implemented. In hindsight, this task was too large for one sprint, this task needed to be broke down into multiple tasks.	This sprint's task was very difficult, time consuming and confusing.	Project Backlog has been reviewed, to check for too large tasks. However, it has been agreed that each task is <u>achievable</u> .

Figure 10 - The Sprint Retrospective meeting from Sprint week 4

## APPENDIX I

Evidence of Agile processes, sprint week 2.

### SPRINT PLANNING

Project Task	Sprint tasks	Estimated Time in Hours	Sprint Backlog	Estimated Time in Hours
Create a button that steps through each hour csv file, which loads each file's ozone data and updates the map live, to display the new ozone figures				
	Create a feature to detect keyboard presses	2	Create a feature to detect keyboard presses	2
	Next and previous ozone 1-hour intervals are loaded upon left or right arrow key presses	3	Be able to load and re-plot ozone layer on map	3
	Be able to load and re-plot ozone layer on map	2	Next and previous ozone 1-hour intervals are loaded upon left or right arrow key presses	2
	Test that each ozone 1-hour interval loads within 0.5 seconds	1	Test that each ozone 1-hour interval loads within 0.5 seconds	1

Figure 11 - The Sprint planning meeting for Sprint week 2, breaking down the Product Backlog feature into smaller tasks for the Sprint backlog

## SPRINT SCRUM MEETINGS

Date	Sprint Week	Sprint Task	Scrum Meeting	Work Achieved Since Last Scrum	Notes, challenges, special info	Current Daily Task
27/02/2020	2	Create a feature to detect keyboard presses	1	New Sprint	First Scrum meeting. Clear understanding of what task to be impleted today.	Create a feature to detect keyboard presses
28/02/2020	2	Be able to load and re-plot ozone layer on map	2	Implemented code that detects keyboard input	There are multiple examples of ways in MatLab to do this however, it took long to find one that would work	Be able to load and re-plot ozone layer on map
29/02/2020	2	Next and previous ozone 1-hour intervals are loaded upon left or right arrow key presses	3	Each ozone 1-hour interval can be loaded and re-plotted onto the European map	Re-used example code to convert the .fig files into .csv files therefore, was not difficult	Next and previous ozone 1-hour intervals are loaded upon left or right arrow key presses
01/03/2020	2	Test that each ozone 1-hour interval loads within 0.5 seconds	4	When right arrow pressed, the next ozone interval is loaded, when left arrow is pressed, the previous ozoneinterval is loaded	No issues, everything was simple to implement	Test that each ozone 1-hour interval loads within 0.5 seconds

Figure 12 - Each SCRUM meeting from the sprint week 2

## SPRINT REVIEW

Date	Sprint Week	Sprint task	Tasks Completed	Product Backlog Updates	Next Sprint Plan Date
02/03/2020	2	Create a button that steps through each hour csv file, which loads each file's ozone data and updates the map live, to display the new ozone figures	Create a feature to detect keyboard presses	A new task has been placed onto the top of the Project Backlog. This is due to the loading of the ozone data fails to meet the 0.5 second execution time requirement. This new task is to modify the code so only the ozone data is re-plotted, and not the entire map	03/03/2020
			Be able to load and re-plot ozone layer on map		
			Next and previous ozone 1-hour intervals are loaded upon left or right arrow key presses		
			Test that each ozone 1-hour interval loads within 0.5 seconds		

Figure 13 - The Sprint Review meeting from Sprint week 2

## SPRINT RETROSPECTIVE

Date	Sprint Week	Sprint task	Self Analysis	Notes, challenges, special info	Improvements
02/03/2020	2	Create a button that steps through each hour csv file, which loads each file's ozone data and updates the map live, to display the new ozone figures	This sprint was overall a success due to the feature being implemented. However, it did not meet the requirement of 0.5 second execution time. Therefore, a new sprint will be needed to address this problem.	No issues however, finding a solution to detecting keyboard input was time consuming. The feature failing the test is not a major issue, due to a solution has already been thought of	The feature will be improved during the next weeks sprint to meet the 0.5 second requirement

Figure 14 -The Sprint Retrospective meeting from Sprint week 2

## APPENDIX J

Mar 1, 2020 – Apr 18, 2020

Contributions: Commits ▼

Contributions to master, excluding merge commits

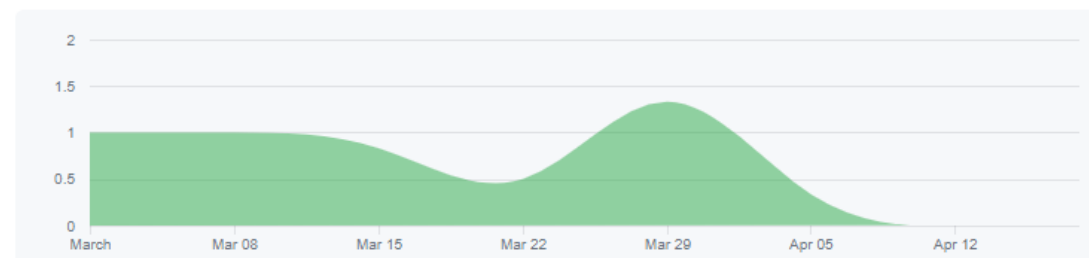


Figure 15 – GitHub contribution overview

## APPENDIX K

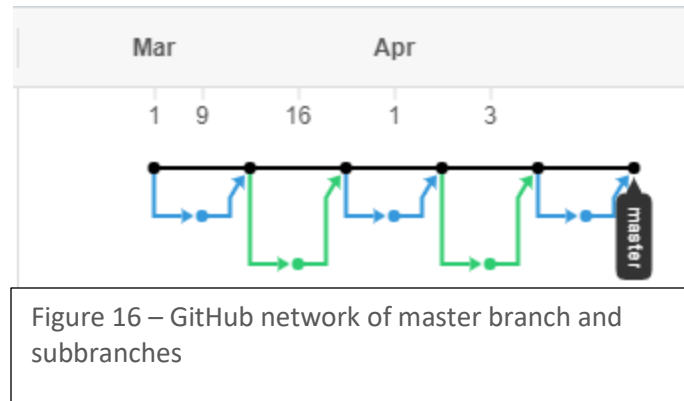


Figure 16 – GitHub network of master branch and subbranches