Project 8 - Buoy Data Generator

Requirements

The Buoy data generator will generate new, randomised data based on the CSV file specifications of existing NDBC Buoy data files which are in CSV file format. The existing NDBC Buoy data files have previously been converted over from NetCDF file format to CSV file format using a separate program called NetCDF\_to\_csv.py which is currently a Python program run via the command-line.

**User Requirements Story:**

A user has existing NDBC buoy data from the year 2020 and wants to generate 7 variations of simulated data the years 2022 through to 2025. The user should:

1. Run/Start the Buoy Data Generator application

NOTE: the program will automatically search for and load config.yaml in the root folder

1. Either input manually or use the up/down arrow keys to show the number 7 in the “instances” field and click the button “apply” to its left.

NOTE: 7 windows will pop up with dialogue and interactable boxes and buttons.

1. Use the checkboxes and radio buttons to enter the environment behaviour desired  
    OR  
   Click the button marked “load as…” then navigate and select the 2020 CSV file from the file manager prompt.

NOTE: this will grey out the radio buttons until you press the “Clear” button besides the “load as…” button.

1. Click ‘Run All’ to run the data generation process based on the selected configuration file  
    OR  
   Click ‘Run’ on each instance window.
2. Navigate to the root folder and the user will now see 7 respective csv files which contain the recently generated data.

**Application Requirements:**

**Main window:**

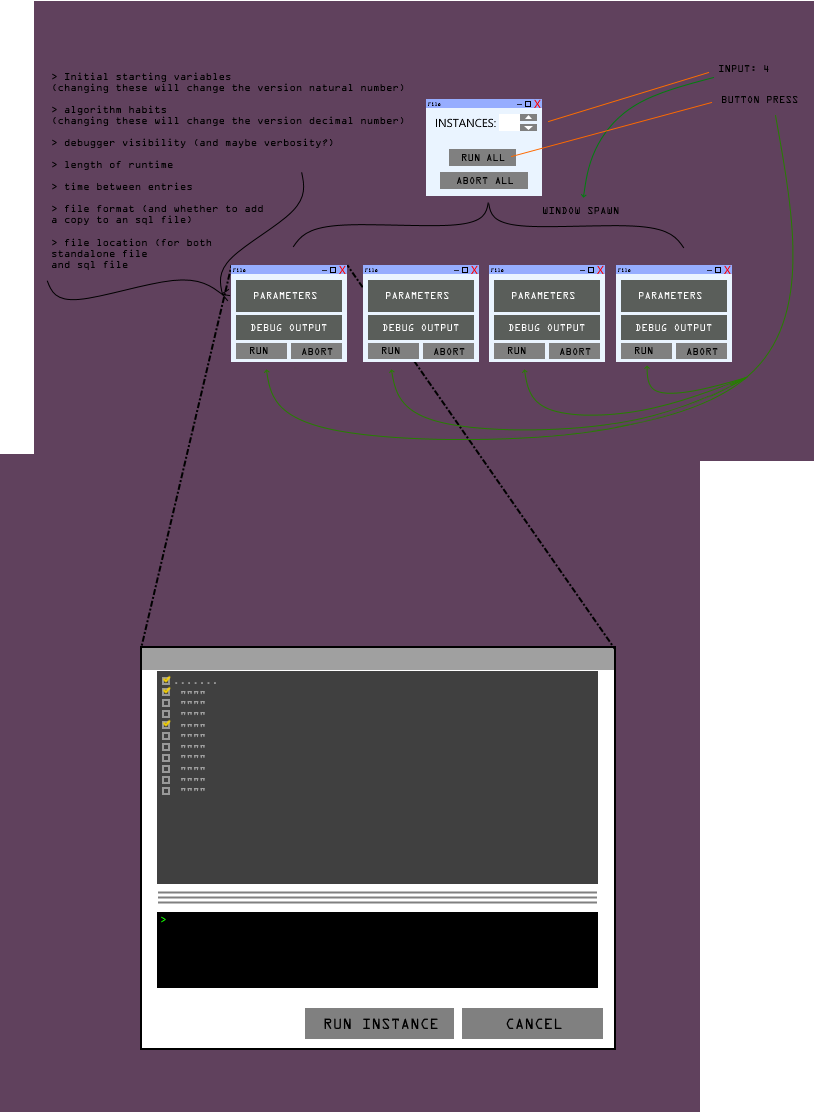
* …Needs a number field with up-down buttons called “Instances”
* Should have a button called “Submit” that validates and ultimately spawns top-level windows called “instances” based on the number inserted.
* …Needs a button called “Run all” which is greyed out when no number greater than 0 is in “instances”
* …Needs another button called “Close all” which is greyed out when no instances have been conceived
* …Should close from use of the in-built “x” in the top-right.

**Instances:**

* …Should have both a config dial dashboard and a file dialogue button called “Load as…”
* “Load as…” should also have an adjacent text field where the selected file name is shown and should be populated with “…” when no file is selected.
* Below, there needs to be a command line output terminal that displays all operations and errors.
* …Needs a button called “Run” that will only run that specific instance.
* …Also needs a button next to “Run” called “Close” that aborts the instance and closes that specific top-down window.

**Config Parameters:**

* …Must directly interact with a .yaml file used for configuration.
* …Should allow basic manipulation of starting variables.
* …Must be greyed out IF there is already a file selected by  
  “Load as…”.
* …Can have a button called “Advanced” that shows additional and more specific settings.
* …Needs to be able to specify what attributes to show in the output file.



**Configuration File Requirements:**

|  |  |
| --- | --- |
| **---** | **DESCRIPTION:** |
|  | Internal applications parameters |
| Theme: | Aesthetic settings section |
| Background: 0x555555 | Background Colour |
| Forground: 0x888888 | Foreground Colour |
| Highlights: 0xaaaaaa | Trim Colour |
| text: 0xdddddd | Text Colour |
| Dimensions: | Window Default Size |
| X: null | Horizontal Plane |
| Y: null | Vertical Plane |
| Position: | Default Location of Window on the Screen |
| X: null | Horizontal Plane |
| Y: null | Vertical Plane |
| Defaults: | Default Config Settings on First Launch |
| Dark theme: True | Major Setting for Theme |
| History: null | Stores IDs and respective values associated with said IDs |
| BASE\_FOLDER: C:/Users/afox/Downloads/cmanwx | Base location of the existing data |
| BASIS\_YEAR: '2020' | Basis data on which to base the generated data |
| BASIS\_MONTH: '01' | Basis data on which to base the generated data |
| CSV\_FOLDER: 'csv' | Input data is in: BASE\_FOLDER/year/month/csv/... |
| MAX\_FILES\_TO\_RUN: 1000 | The maximum number of files/buoys to generate data for |
| OUTPUT\_YEARS: |  |
| [2022, 2023, 2024, 2025] | List of the years desired for the output sim data |
| OUTPUT\_MONTHS: |  |
| [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] | List of the months desired for the output sim data |
| DataFrequency: 1 | Desired frequency of the generated data in hours |
| MonthLength: |  |
| [0,31,28,31,30,31,30,31,31,30,31,30,31] | Constant Defining the standard month lengths |
| Statscolumns: | Statistics to be calculated from existing data |
| - FieldName | Name of the field |
| - DataType | Type of the field (int, float, string, datetime) |
| - Min | Numerical minimum |
| - Max | Numerical maximum |
| - Mean | Numerical average (mean) |
| - StdDev | Standard deviation |
| - Median | Field median value |
| - Mode | Most likely/common value of this field |
| - NumValues | Total number of values (including nulls) |
| - NumNulls | Number of null values |
| - NumUnique | Number of unique values for this field |
| - AutoCorr | Field self-correlation |
| - FFT | Data periodicity (hrs) |
| - Distrib | Distribution (discrete only) |