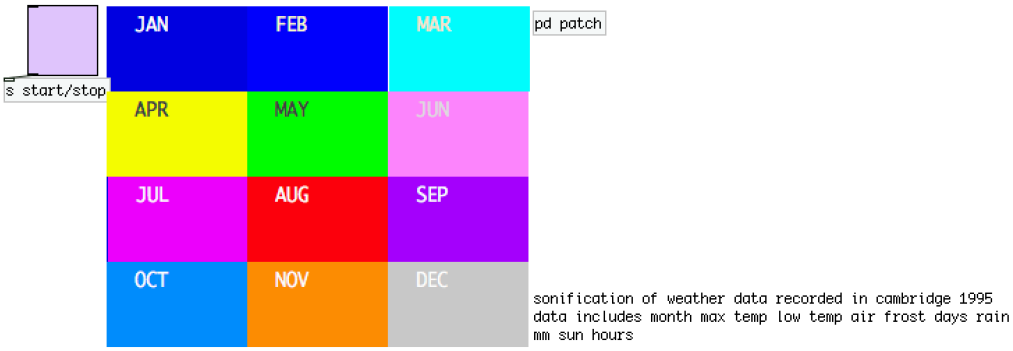
The sonification of weather data using Pd-Extended

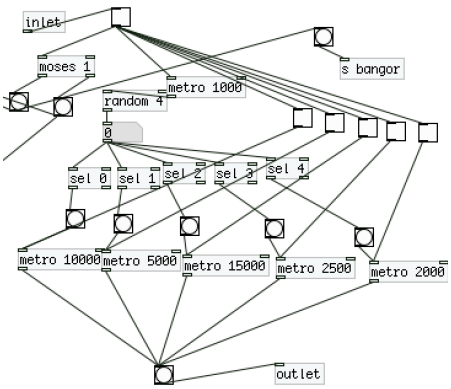


To begin the process of turning information into audio, formatted data is needed, for this project data was taken from the met office and represents weather readings taken in Cambridge,UK in the year1995.

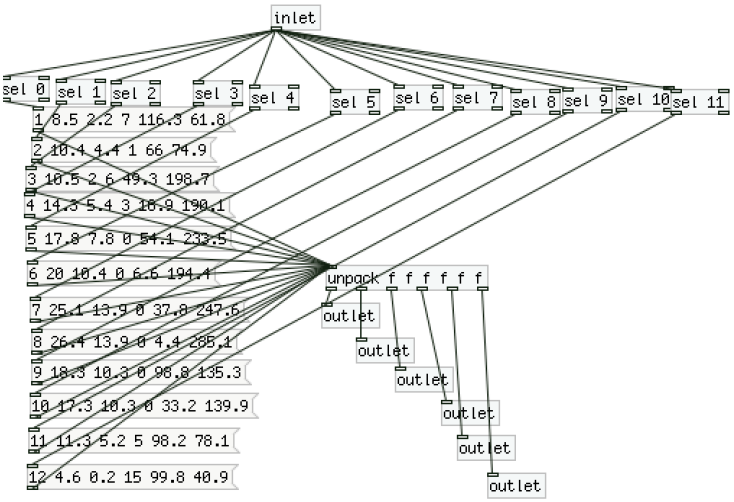
yyyy	mm	tmax degC	tmin degC	af days	rain mm	sun hours
19951	8.5	2.2	7	116.3	61.8	
1995	2	10.4	4.4	1	66.0	74.9
1995	3	10.5	2.0	6	49.3	198.7
1995	4	14.3	5.4	3	18.9	190.1
1995	5	17.8	7.8	0	54.1	233.5
1995	6	20.0	10.4	0	6.6	194.4
1995	7	25.1	13.9	0	37.8	247.6
1995	8	26.4	13.9	0	4.4	285.1
1995	9	18.3	10.3	0	98.8	135.3
1995	10	17.3	10.3	0	33.2	139.9
1995	11	11.3	5.2	5	98.2	78.1
1995	12	4.6	0.2	15	99.8	40.9

As shown in the image the data includes; year, month, max temp in degC, min temp in degC, air frost days, rain in mm and sun hours. More information would likely mean there would be a significant amount of different factors being signalled during the patch, resulting in a possible lack of clarity. It may be considered important that each piece of information is recognisable within the A/V piece.

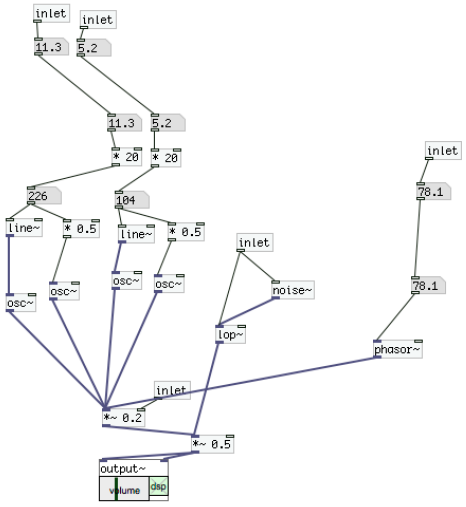
The patch begins with the metronome, to make the piece rhythmically more appealing and less receptive the metronome makes use of the PD random object. A metronome set at 1000ms triggers a random number between 0 and 4, the number then selects the speed for that period. The use of a random metronome was used to inflict the piece creatively however another option could be to allow a piece of data to affect the metronome.



The next step was to implement the data into the patch in order for pure data to read it. Using the message function allowed the months to be separated into categories, using the unpack object, the data is stripped into individual chunks as long as it is separated correctly. Like the metronome the data is chosen at random, meaning any month can come in any order, this is to allow the data to represent itself in an audible way, rather than a linear approach which is easier to predict. once a random message is banged, the unpack outputs the 6 different number values.



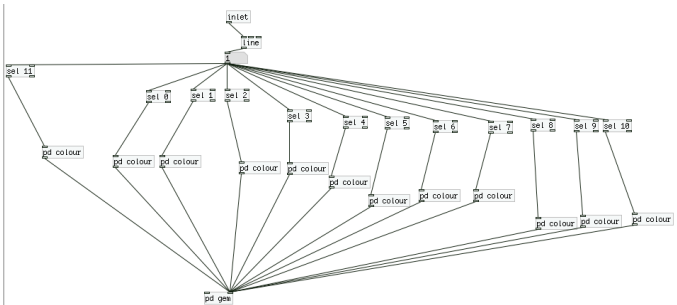
Once the data has been inputted the sonifcation process can begin, the patch makes use of both sound and vision to represent different pieces of data. It is important to map out what the data is to represent in order to make use of all the information.



Both max and min temperature control two sine waves, the input signal is multiplied by twenty in order for the data to be within an audible frequency. a lower frequency would indicate colder weather, a higher frequency means it was warmer that month, the larger the frequency gap between the two would indicate a rapidly changing month, for example May has a maximum temperature of 17.8, and a minimum of 7.8.

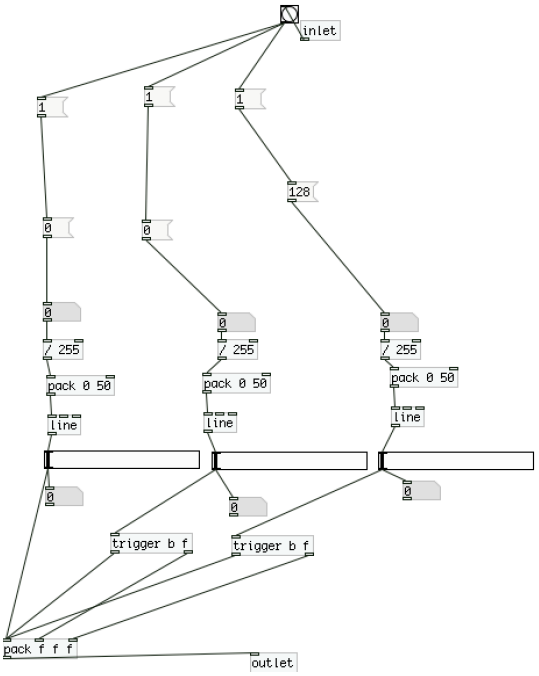
Rain fall is represented by a Noise ~ object and a low pass filter, the higher the amount of rainfall, the more open the filter becomes, creating an audible signal of rain.

Sun hours are represented with a brighter saw wave, this is to sonically separate this information and previous data, the more sun hours within a month the higher pitched the saw wave will be.



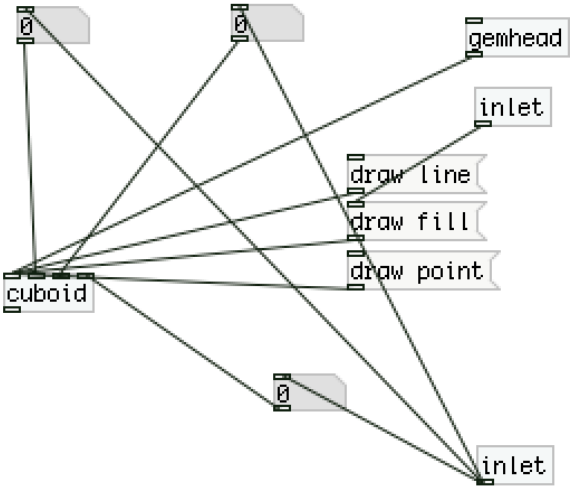
The selected month is represented with colour making use of PD's Gem environment.

The selected month triggers a specific RGB value, allowing the colour in the gem



window to correspond with the correct month being sonified.

the RGB sub sends information to



the Gem window.

Air frost data is also represented visual. if present a white cube shape will appear in the gem window, when air frost doesn't occur in that month the white cube will not appear.

Adding all previous elements together allows the metronome to trigger and stop an audio visual display of data. Using send and receive objects the patch is neatly wrapped into a sub patch. A GUI is created to allow the user to start/stop the patch and to also see the correlation between colour and month.