## Arts Application Programming Lab Exercises: Week 5, Visualisation & Colour

|  |  |  |
| --- | --- | --- |
| **Lab5\_1\_colour** | Aim: understand and use colours | **Grade: F (E for hex)** |

Using fill() and stroke() and various shapes experiment with colour (try not to use the colour picker)

Try to achieve at least the following colours:

* yellow, cyan, magenta (the secondary colours)
* orange
* purple
* pink
* light green
* dark blue

Note, from the Processing colour tutorial (https://processing.org/tutorials/color/): “While this may take some getting used to, the more you program and experiment with RGB color, the more it will become instinctive, much like swirling colors with your fingers.”

Repeat this with transparency (e.g. draw overlapping rectangles) and both the hexadecimal notations: #AABBCC and 0xAABBCCDD

Then change the colour mode to HSB and explore this (aim: to see what might help most with your coursework visualisation)

Finally use the colour picker for more subtle colours

|  |  |  |
| --- | --- | --- |
| **Lab5\_2:** | Aim: basic visualisation using 1-d and 2-d arrays | **Grade: C** |

Write a sketch that draws a bar chart for a set of data representing, e.g., average rainfall in each month of the year for a city/country of your choice (search to find appropriate data):

* Create a global array that contains the 12 values
* In draw call a drawData() function
* Define a drawData function that draws the bar charts

a) simplest: draw the bar charts going down the page

b) draw them coming up from the bottom (as you would normally see a bar chart)

c) draw them going horizontally from the left of the window.

Harder (optional – complete the other exercises before exploring 2D arrays): Extend your sketch by using a 2-d array to record different shades of grey for each of the vertical bars.

Optional (to explore at home): in the 2nd edition of the Processing book example 12-2 uses beginShape(), vertex() and endShape(). The purpose of this module is not to train you in as many ready-built Processing functions as possible, but this could be useful for your final visualisation depending on your data and chosen story.

|  |  |  |
| --- | --- | --- |
| **Lab5\_3:** | Aim: practice reading in data from CSV files | **Grade: B** |

A CSV file is a text file representing a spreadsheet consisting of columns of values separated by commas. Each row, terminated by a line ending, represents a data item & the columns the different properties of that data item.

1. As an example, create a text file called ‘numbers.csv’ with the following 8 lines:

735,465,202,143,390,260,291,470,332,410,338,104,375,377,538

320,600,329,890,250,230,710,680,248,460,981,735,170,425,100

325,330,290,350,327,360,340,570,111,260,130,233,545,276,105

392,336,100,407,230,730,217,350,370,255,500,435,445,210,314

524,280,600,286,265,600,281,300,215,431,696,490,185,800,237

450,315,505,280,850,306,352,560,253,350,605,310,380,420,610

145,870,145,525,210,390,259,900,780,760,250,460,300,350,360

135,906,840,720,630,420,250,625,280,618,411,383,380,360,540

a) Using loadTable(…) read in this data into a Table object. Then printout the contents of the table.

b) Write a function to display the data. Use the first number in each line to correspond to the R value in fill, the second as G, the third as B and the last as alpha. Hint: use the Processing map() function to constrain the CSV values to appropriate values for colour. Use the next four numbers as parameters for ellipses. The next three should correspond to the outline colour.

After finishing this lab, ensure you have finished the labs from week 4 (and if so, completed the games)