Creative Coding 1 CA

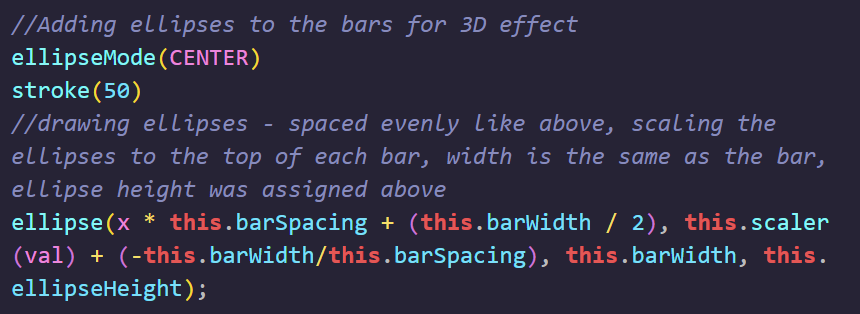
Liam Ronan

N00212101

1. **3D effect on bars using ellipses**

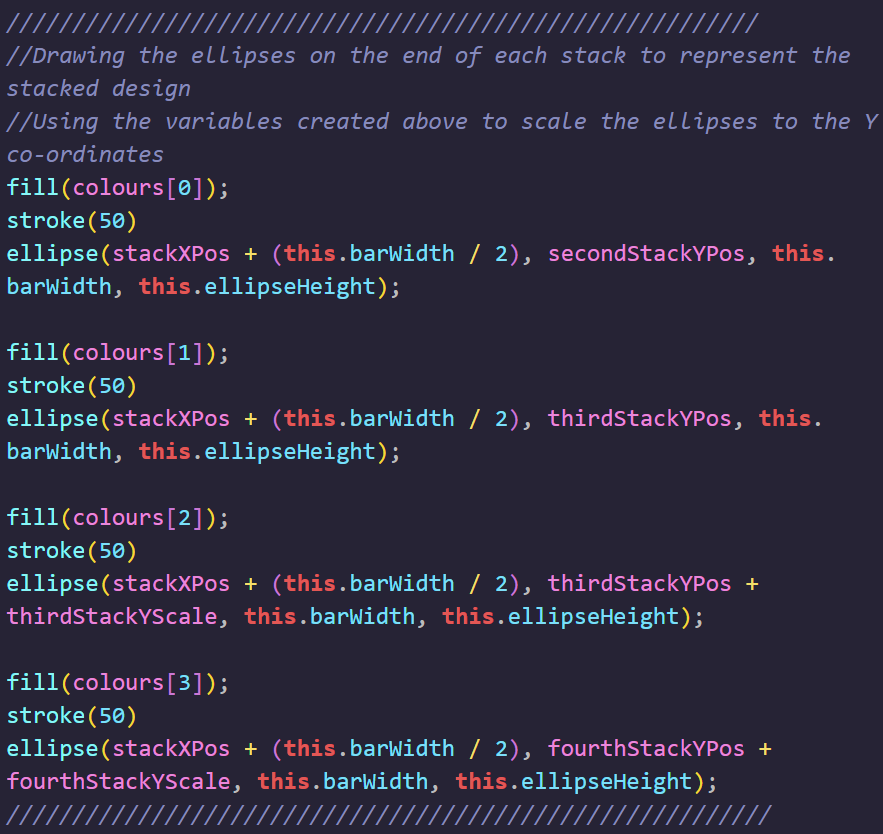
I used ellipses on all regular bar charts in order to create a 3D effect. To do this I mostly added on to the drawBars() method by using the same variable I used within the for loop to generate each bar. By doing this, I can easily set the x co-ordinates on each ellipse. The Y co-ordinates is simply the scale value of each bar plus the bar width divided by the bar spacing. The width of each ellipse is the same as the bar width, and the ellipse height was declared and initialized above to 5.

Below is a screenshot of the code for the ellipses on a regular bar chart:



Figure

In regards to the horizontal stacked bar chart, because of how each stack adds onto the last, the same had to be done for the ellipses. Also, showing the ellipses on the horizontal stacked bar chart was beneficial because it differentiated the stacks. The same as above was done for the ellipses at first but in terms of conciseness and readability, I decided to convert this particular chunk of code into the drawStacks() method as seen below:



Figure

1. **Scatter Plot data points**

To draw the points on the scatter plot chart, I was required to scale from the X-axis and Y-axis. In terms of how I did this was by using the p5 map method. I created scaleXData and scaleYData methods which gets passed a value for the ellipse X co-ordinates and Y co-ordinates.

Fig 3 below shows the scaleXData() method and fig 4 shows the ellipse with the returned value from the method stated above:

A picture containing diagram

Description automatically generated

Figure

Text

Description automatically generated

Figure

1. **Scatter Plot Data Importance**

Due to the large quantity of data used for the scatter plot chart comparing covid deaths per million VS GDP per capita, I decided to only show the data that was easily readable, had obvious outliers and easily viewed differences between data.

To achieve this, in the app.js I specify true when calling the render method, next I pass a Boolean variable called \_setLabel to the render method with a default of false in the parameters. This then will be passed into the drawPoints() method where I draw the ellipses. DrawPoints() will be passed the variable and an if statement will run where I draw the text above each ellipse, basically defining that if \_setLabel is true, and if the data array 1st item is greater than the max number divided by then draw the text.

Here is the code below in *figure 5:*

Text

Description automatically generated

Figure

1. **JavaScript reverse method**

Due to the way I drew the bars within the horizontal bar chart class, the Y-Axis labels and actual bar values were opposites. For getting the text on the why Y-Axis, I was using the getColumn method and passing in the value was passed into the constructor as seen below:



Figure

When I wanted to use the columns for my Y-Axis labels, I declared another variable in the drawYAxis method and reversed the array shown above. Then I looped through it the same as before to get my values but in the reversed order to line up with my bars.

Text

Description automatically generated

Figure

1. **Colour Array**

For the colours of the bars and points on the scatter plot, I created a colours array in the setup function in the app.js.

Text

Description automatically generated

Figure

To use the colours array throughout the various classes, simply used fill(colours[x & colours.length]). The x references the for loop that would be drawing the bars or the points of the scatter plot.

Text

Description automatically generated

Figure

Also, for my stacked bar chart to keep the first value ie. Male, to be all the same colour, I received only the first colour of the array such as this:



Figure

Then the same as above for the second or third stack.