

Impacts of Covid-19 on Irish Airports



**Nathan Moore**

**N00201743**

**GitHub Link:** <https://github.com/nathanmoore21/CreativeCoding2/tree/master>

Creative Coding 2

CA 1 – Building a Chart Classes in JavaScript

Year 2 2021-22

DL836 BSc (Hons) in Creative Computing

# Approach Taken in this CA

We started off our Creative Coding Two classes learning how to create different types of graphs. Starting off with bar charts we got to learn the fundamentals of creating charts using p5.js. Like any JavaScript class, it can be a little overwhelming and tricky to get your head around.

We worked on in-class tasks to improve our knowledge so we could eventually start our CA. For my CA, I decided to do the impact of the pandemic on travelling overseas as it is something that affected nearly everyone and is also an interesting story to tell. After doing some research I discovered three interesting sets of data that all relate to overseas travel.

# JavaScript

In this CA for Creative Coding 2, our individual aim was to create three separate graphs to provide an interesting story. For my graph story, I decided to display three different statistics about how the pandemic has affected overseas travel and the impact it had on main Irish airports.

We were using Visual Studio Code to implement our Java. JavaScript is the most popular programming language that allows you to create complex, sometimes interactive features on web pages. P5.js is a JavaScript language, it’s aim is make coding more accessible for designers, programmers etc.

# Conditional Statements

Conditional statements can be very beneficial in programming languages such as JavaScript. Programmers all over the world already use JavaScript as it is a very useful programming language for customization. The ‘if’ statement is one of the conditional statements, it allows you to perform different actions depending on the decision. An example of the ‘if’ statement that I used in my CA is;

if (this.rotateLabels) {

push()

noStroke();

textSize(14);

textAlign(LEFT, CENTER);

translate(((this.barWidth + this.BarSpacing) \* i) + this.barWidth/2, 10);

rotate(PI / 2.5)

text(this.data[i].name, 0, 0);

pop()

} else {

noStroke();

fill(255);

textSize(14);

textAlign(CENTER, BOTTOM);

text(this.data[i].name, ((this.barWidth + this.BarSpacing) \* i) + this.barWidth / 2, 20);

In this example, you can see that if rotateLables is set to true the following will be displayed on the page. If not, the ‘else’ statement will be placed on the page.

# Modulus Operator

Another useful feature in Java is the use of the modulus operator (%) The modulus operator returns the remainder of two numbers after division. In the example below you can see that 4 goes into 10 twice with a remainder of 2. This can be beneficial for many reasons, in my CA I used it to loop through code so each bar on the bar charts is different colours which makes it easier for readability.

Let colorNumber = I % 4;

# Boolean Data Type

The Boolean data type is a type of data that can only have one of two possible answers – true or false. It’s beneficial for programmers to program code to give a true/false, yes/no answer. An example of this that I’ve used in my CA would be:

If I have the following code set to true display;

if (this.showValues) {

fill(255, 200);

noStroke();

textSize(12);

textAlign(RIGHT, CENTER);

text((i \* this.tickIncrements).toFixed(this.numPlaces), -this.tickSpacing \* -i, 20);}

If not, no values will be displayed. I have my showValues set to true as I would like the figures for each chart to be displayed.

this.showValues = true;

I originally had my showValues set to false as my charts were looking too messy and there was too much going on. After I organised my data and code more having my data values displayed looked much better and it was only a matter of changing my ‘false’ to a ‘true’.

# Translations

In programming, when you create a canvas. The origin point is automatically (0,0) on the x-axis and y-axis. This is located in the top left-hand corner of the screen, so everything is automatically drawn from that point. This can become very un-useful if there is a lot being programmed or in my case, creating graphs. Using translations can help you create a new origin point rather than the original (0,0). Translations are very helpful if you intend on relocating the graphs. In my CA, I used translations for each graph, this line of code was used to locate the labels of each bar.

translate(((this.barWidth + this.BarSpacing) \* i) + this.barWidth / 2, 10);