Lifespans In Middle Earth Data Art - Report

**Final Results**

A picture containing circle, colorfulness, light, spiral

Description automatically generatedThe final Data Art piece, named Lifespans In Middle Earth, successfully portrays the lifespans of various characters. It does this by displaying each character as a circular arc, each with a unique colour for satisfying distinguishment, the colours were mostly chosen based on the association with the given character. Each arc has a white ellipse that stays with the end of the arc, indicating the end of a character’s life.

A white line is generated from the centre of the piece that protrudes upwards to the edge of the coloured arcs, referencing the average age of death for a character’s race. When a character’s circular arc extends past the white line, that means they lived past the average lifespan of their race, and it can be seen by how much on the piece.

Also included, is the names of each character labelling their respective circles, the letters are coded to curve along with the circular arcs. Upon running the code, the art piece will iteratively be drawn over a period of time, giving the piece a dynamic dimension for potential online viewing.

To enhance the Tolkien theme the outside of the circular arcs shows an image of the inscription from the One Ring, written in the fiery letters of Black Speech, this image was sketched in Photoshop and imported into Processing. This piece was also inspired by the iconography of the Eye of Sauron; hence the art piece appears eye shaped and similarly ‘wreathed in flame’ in a way.

**Techniques Used**

To create the Data Art piece, many methods were employed:

**Data processing:**

Before the data could be used, it had to be properly adapted for the processing scripts. This was done by calculating the age of death for each of the selected characters from the dataset, this had to be done since the original dataset did not have values for age of death. This was completed by subtracting the character’s year of death from their year of birth, although this couldn’t be done for all characters due to the changeover of the Third Age to the Fourth Age of Middle Earth, when the year would change from 3021 TA (Third Age) to 1 FO (Fourth Age), this was rectified through use of this simple formula:

A picture containing text, handwriting, calligraphy, ink

Description automatically generated

Once the age of death had been determined for each character, some more data that was not included in the original dataset had to be retrieved, that being the average age of death for each race, this data was obtained through internet sources such as [LOTR Fanon - Fandom](http://lotrfanon.fandom.com/) and [Tolkien Gateway](http://tolkiengateway.net/). Once the all the data was collected it was organised into a separate spreadsheet:

A picture containing text, number, screenshot

Description automatically generated

The angle quadrants in Processing had an offset of 90° so the positive y-axis started at -90° instead of 0°, to account for this when calculating the necessary angles for each character, the formula of θ° = ( (360 \* (x / y) ) – 90) was used where x is the age of death, and y is the average lifespan, the subsequent values would be ready to use within Processing.

The characters were also sorted highest to lowest based on their screen-time in the films.

A colour palette was also selected in the spreadsheet, signifying what the colours of the character circles would be, labelled with their RGB number for simple transfer to Processing.

**Arcs:**

The primary shapes used in the art piece were arcs, in Processing, the arc() function was used to accomplish this. The angles made in the spreadsheet were assigned to the *character\_name*End variables since this is the angle where the arc will stop. The colours for the arcs were carried over from the spreadsheet’s RGB values and assigned to the *character\_name*R, *character\_name*G, *character\_name*B variables respectively. To arrange the arcs so that the more prominent characters appear on the outside, the individual arcs were scaled linearly.

A screenshot of a computer program

Description automatically generated with low confidence

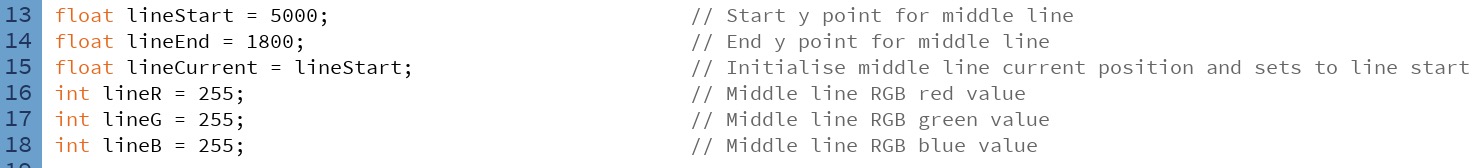
A picture containing text, screenshot

Description automatically generated

A picture containing text, screenshot, algebra

Description automatically generated

**White Line:**

The central white line was coded similarly to the arcs, using lineStart and lineEnd variables to define where it should begin and finish. 

**Animated Generation:**

This art piece is able to generate iteratively over a span of time through the use of the draw() functionality, as well as if() statements encompassing the draw commands. The if() statement condition would be – if(*character\_name*Current is less than *character\_name*End) then the shapes, whether an arc or line, would be drawn from the *character\_name*Start position to the *character\_name*Current, which would begin equal to the *character\_name*Start variable, but would then increment by one after each draw, therefore, each time the draw() function is run, the shape would be become more complete, until it reaches the *character\_name*End value.

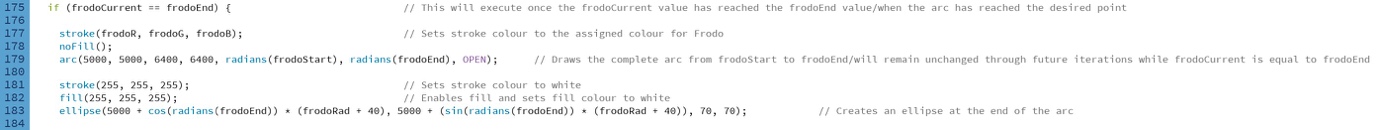
A picture containing text, screenshot, receipt, algebra

Description automatically generated

A picture containing text, screenshot, font, number

Description automatically generated

Once this would happen the entire shape would disappear due to the condition turning false, to fix this, a different if() statement was introduced with the condition of if(*character\_name*Current is equal to *character\_name*End), it would then draw the complete shape, since this condition will always be true as long as the image is complete, the image will remain.

****

To enhance the visual aesthetic, as well as to aid the viewer in distinguishing the end points for the lines, a white ellipse was added to the end of each character line, the ellipse follows the same animation as the line and remains at the end of it through the use of a mathematical function that can calculate a point on a circle, given the circle radius and current angle: x = r(cos(θ‎°)) and y = r(sin(θ‎°)).

****

To retain the image of the ellipse when the image is fully generated, the ellipse is drawn again in the new if() statement similarly to the arc:

****

**Curved Text:**

One of the most challenging aspects of the art piece was to be able to curve the names of the characters with the arcs. To accomplish this, I found some code [online](http://learningprocessing.com/examples/chp17/example-17-08-textalongcurve) and adapted it to work for each of the arcs in the art piece. The method consists of a loop that takes each letter of a given string and applies a transform to the individual letters, giving the effect that a word is curving. The letters would be transformed in reference to the radius of the arc, and the output word would appear to have the same curvature as the arc. For each character, the text position on the circumference had to be tweaked manually.

****

**A picture containing text, screenshot, receipt

Description automatically generated**

**Black Speech Image:**

The outer image of the Black Speech inscription that appears on the One Ring was sketched in photoshop by tracing over the official image and adding some gradient colour. The image was then curved in a ring shape so that it could surround the art piece, to give reference to the iconic One Ring, and also to the Eye of Sauron.

**A picture containing circle, rope

Description automatically generated**

**Critical Reflection**

Overall, the Data Art piece has achieved the goal of presenting the lifespans of characters of Middle Earth. The data is clearly laid out with little clutter or overuse of colour, the colours themselves feel relevant to the theme. Once understood what is represented in the piece, the data can be clearly understood by the viewer.

Some aspects that could have been better would be the use of less monochrome colours per arc, for example, the colour could be enhanced using textures, to add more detail and intrigue to them. Another aspect to consider improving on would be the incorporation of more data, since the data presented in this piece is mostly one dimensional and does not encourage longer viewing, since the data can be read fairly quickly. A final element that could be better would be the use of more characters, since this would give more viewable data and hence make the piece more appealing.