William Roberts

10017537@napier.live.ac.uk

inLineD Hand over

Contents

[Introduction 3](#_Toc490128292)

[List of Original Files 3](#_Toc490128293)

[List of new files created 3](#_Toc490128294)

[List of changes made to original files 3](#_Toc490128295)

[Functionality added to inLineD 4](#_Toc490128296)

[Rows 4](#_Toc490128297)

[Cols 4](#_Toc490128298)

[Reorder 4](#_Toc490128299)

[Change Guides 4](#_Toc490128300)

[Order Force 4](#_Toc490128301)

[Additional Features 4](#_Toc490128302)

[List of functions created for program. 5](#_Toc490128303)

[EditSvg.html 5](#_Toc490128304)

[Init() 5](#_Toc490128305)

[Rows() 5](#_Toc490128306)

[Cols() 5](#_Toc490128307)

[Reorder() 5](#_Toc490128308)

[Background() 5](#_Toc490128309)

[None() 5](#_Toc490128310)

[ChangeGuides() 5](#_Toc490128311)

[OrderForce() 5](#_Toc490128312)

[LabelListBox() 6](#_Toc490128313)

[AddToPrio() 6](#_Toc490128314)

[PrioList() 6](#_Toc490128315)

[ErasePrio() 6](#_Toc490128316)

[SvgTools.js 6](#_Toc490128317)

[splitSVG() 6](#_Toc490128318)

[RebuildLinesArray() 6](#_Toc490128319)

[RebuildSVG() 7](#_Toc490128320)

[GuidesBackground() 7](#_Toc490128321)

[GuidesNone() 7](#_Toc490128322)

[SortForce() 7](#_Toc490128323)

[SortDForce() 7](#_Toc490128324)

[AddDPlusL() 8](#_Toc490128325)

[AddDPlus() 8](#_Toc490128326)

[AddD() 8](#_Toc490128327)

[FindPresent() 9](#_Toc490128328)

[SortD 9](#_Toc490128329)

[OrderGaps() 9](#_Toc490128330)

[CheckGapsL() 10](#_Toc490128331)

[CheckGaps() 10](#_Toc490128332)

[SwapCols() 10](#_Toc490128333)

[GetXLineEnd() 10](#_Toc490128334)

[GetXLineStart() 11](#_Toc490128335)

[GetYAxisSpace() 11](#_Toc490128336)

[GetYStartLine() 11](#_Toc490128337)

[GetYstart() 11](#_Toc490128338)

[ReDrawOverlaps() 11](#_Toc490128339)

[ChangeGuides() 12](#_Toc490128340)

[Transpose() 12](#_Toc490128341)

[GetLabels() 13](#_Toc490128342)

[GetXPos() 13](#_Toc490128343)

[GetStrokes() 13](#_Toc490128344)

[RowPop() 13](#_Toc490128345)

[GetCols() 14](#_Toc490128346)

[SwapRows() 14](#_Toc490128347)

[GetLines() 14](#_Toc490128348)

[SeparateSVG() 14](#_Toc490128349)

[GetLineNo() 15](#_Toc490128350)

[GetHeight() 15](#_Toc490128351)

# Introduction

This document will inform the reader on changes made to the original inLineD files provided at the start of the project. Areas covered in the following sections will include:

* List of original files.
* List of new files created.
* List of changes made to original files.
* Functionality added to inLineD:
  + List of inputs used.
  + List of function outputs.
* List of functions used and purpose.

# List of Original Files

* Index.html.
  + Program launches from this page.
  + Used to navigate to SVG edit page.
* Utils.js.
  + Holds functions used to create initial SVG image string.
* Linear.html.
  + Draws SVG image.
* FileSaver.js.
  + Used to save SVG string as a txt file.
* Index.css (empty).
  + Unused.

# List of new files created

* editSVG.html
  + Used as user interface for the manipulation of the SVG string and image.
* svgTools.js
  + JavaScript file which contains the functions used to manipulate the svg image.

# List of changes made to original files

* Linear.html updated to put the following into local storage:
  + Svg string
  + Empty array to store prioritised labels
  + Empty array to store prioritised label row
  + Sets the guides variable as ‘guides’
    - (allows the image to be drawn on the next page.)
* Index.html updated to remove guides:
  + Line 47 comment out to stop programming changing guides (guides moved to editSVG.html)
* Lines 83 – 96 commented out to stop user being able to change guides

# Functionality added to inLineD

## Rows

This function will allow the user to input 2 row numbers. The program will then swap these 2 rows.

* Uses 2 integer inputs.
* Will output edited svg string to local storage.

## Cols

This function will allow the user to input 2 column numbers. The program will then swap these 2 columns.

* Uses two integer inputs.
* Will output edited svg string to local storage.
* Will take column spacing into consideration when swapping.

## Reorder

This function will reorder the diagram so that it contains the least amount of line spaces.

* Requires no user input.
* Outputs new svg string to local storage

## Change Guides

This function allows the user to change the guide lines that appear on the diagram. Options available are:

* Lines (default)
* Background
* None

User interaction with this function includes:

* Input from radio boxes with a submit button.
* Outputs new svg string to local storage.

## Order Force

This function allows the user to take a list of available labels and place selected labels into a priority list. From here, the program will redraw the diagram while putting the least amount of spaces in the specified line. Uses recursion to reorder lists with more than 1 priority label.

* Input from an array of priority labels.
* Outputs new svg string to local storage.
* Also has option to erase priority list.

## Additional Features

The following features have also been added:

* Download option to allow user to download updated image.
* Option to go back to index.html page (this will allow the user to input new diagram data).

# List of functions created for program.

## EditSvg.html

### Init()

Runs when page is loaded or refreshed. Checks for guide status in local storage and draws diagram, label lists and priority lists.

### Rows()

Function called by form on editSVG.html. Calls swapRows function on svgTools.js and passes 2 integers.

**Input**: Integer x 2

**Output**: N/A

### Cols()

Function called by form on editSVG.html. Calls swapCols function on svgTools.js and passes 2 integers.

**Input**: Integer x 2

**Output**: N/A

### Reorder()

Function called by form on editSVG.html. Calls orderGaps function on svgTools.js.

**Input**: N/A

**Output**: N/A

### Background()

Function called by form on editSVG.html. Calls guidesBackground function on svgTools.js.

**Input**: N/A

**Output**: N/A

### None()

Function called by form on editSVG.html. Calls guidesNone function on svgTools.js.

**Input**: N/A

**Output**: N/A

### ChangeGuides()

Function will check to see which radio box is ticked and change the ‘guides’ value in local storage accordingly. Used for changing diagram guides.

**Input**: Radio box selection

**Output**: Updates ‘guides’ in local storage.

### OrderForce()

Function used to reorder diagram based on priority lines. Calls sortForce Function from svgTools.js and passes priority number array.

**Input**: svg string from local storage, priority line array, priority line number array.

**Output**: N/A

### LabelListBox()

Function used to create and populate a list box with all available labels.

**Input**: svg string from local storage.

**Output**: string containing html code for form with select options.

### AddToPrio()

Function used to move a selected label to a priority list array.

**Inputs**: svg string from local storage

Existing priority labels from local storage.

Existing priority label numbers from local storage.

**Outputs**: Sets new priority labels array to local storage.

Sets new priority label number array to local storage.

### PrioList()

Creates and populates a priority list box.

**Inputs**: svg string from local storage.

Priority labels from local storage.

**Outputs**: string containing html code for form with priority labels.

### ErasePrio()

Function erases the priority labels list.

**Inputs**: priority labels from local storage.

Priority label numbers from local storage.

**Outputs**: Sets empty priority labels array in local storage.

Sets empty priority label numbers array in local storage.

## SvgTools.js

### splitSVG()

**Lines:** 1-8

Function used to split SVG string into an array of lines. Splits the string by “<” character. This character is added when the string is rebuilt.

**Input**: svg string passed in as variable.

**Output**: returns array of individual lines.

### RebuildLinesArray()

**Lines:** 9-19

Function used to rebuild array of lines from individual sections.

**Input**: Array of sections.

**Output**: Array of Lines.

### RebuildSVG()

**Lines:** 20-31

Function used to turn an array of lines back into the complete svg string. Adds the “<” character to the start of everyline.

**Input**: Array of individual lines.

**Output**: Svg string.

### GuidesBackground()

**Lines:** 32-99

Function used to change the guides on the svg image to background rectangles. This edits sections[0] to change the start of the svg string.

**Input**: svg string from local storage.

**Output**: returns new svg string.

### GuidesNone()

**Lines:** 100-110

Function used to remove guidelines from the svg image. This edits sections[0] to change the start of the svg string.

**Input**: svg string from local storage.

**Output**: returns new svg string.

### SortForce()

**Lines:** 111-208

Function used to sort svg diagram to allow for the minimum number of total spaces between all lines.

**Input**: Array of priority labels passed in as variable.

Svg string from local storage.

**Output**: calls redrawOverLaps() function to put new svg string in local storage.

### SortDForce()

**Lines:** 209-347

Function used to sort the svg diagram based on a prioritised line. Uses recursion to split and rebuild sections of the diagram when there is more than one priority label. Note: input and output required to be the same for recursion to work properly.

**Input**: Array containing the following arrays

* Array of names.
* Array of overlaps (diagram).
* Array of column spacing.
* Array of priority numbers (line No. on diagram).

**Output**: Array containing the following arrays

* Array of names.
* Array of overlaps (diagram).
* Array of column spacing.
* Array of priority numbers (line No. on diagram).

### AddDPlusL()

**Lines:** 348-415

Function used to add 2 diagram sections together finding the least number of gaps in a specified line.

**Input**:

* Diagram 1.
* Diagram 2.
* Array of spaces for Diagram 1.
* Array of spacers for Diagram 2.
* Prioritised line number (integer).

**Output**: Array containing:

* Diagram.
* Column spaces for diagram.

### AddDPlus()

**Lines:** 416-475

Function used to add 2 diagram sections together with the least number of total line spaces.

**Input**:

* Diagram 1.
* Diagram 2.
* Array of spaces for diagram 1.
* Array of spaces for diagram 2.

**Output**: Array containing:

* Diagram.
* Column spaces for diagram.

### AddD()

**Lines:** 476-491

Function used to add 2 diagram sections together (crude method, doesn’t check for spaces, used by AddDPlus() and AddDPlusL()).

**Input**:

* Diagram 1.
* Diagram 2.
* Array of spaces for diagram 1.
* Array of spaces for diagram 2.

**Output**: Array containing:

* Diagram.
* Column spaces for diagram.

### FindPresent()

**Lines:** 492-539

Function used to determine split a diagram into 2 sections. One section contains all columns with a specified line present, the other contains all columns with the specified line absent.

**Input**:

* Array of columns.
* Array of spaces for columns.
* Prioritised line number.

**Output**: Array containing the following arrays

* Array of present columns.
* Array of spacing for present columns.
* Array of absent columns.
* Array of spacing for absent columns.

### SortD

**Lines:** 540-591

Function used to sort all columns to give the least number of total gaps in lines.

**Input**:

* Array of diagram rows.
* Array of spacing for each column.

**Output**: Array containing:

* Updated Array of diagram rows.
* Updated Array of spacing for each column.

### OrderGaps()

**Lines:** 592-674

Function called by editSVG.html. Used to reorder diagram to provide minimum number of total gaps in lines. Uses SortD() function.

**Input**: svg string from local storage.

**Output**: Calls redrawOverLaps() function to put updated svg string in local storage.

### CheckGapsL()

**Lines:** 675-697

Function used to find the number of line gaps in a specified line.

**Input**:

* Array of Rows
* Priority line (Integer)

**Output**:

* Number of gaps in a line (Integer)

### CheckGaps()

**Lines:** 698-731

Function used to find the number of line gaps for the entire diagram.

**Input**:

* Array of rows.

**Output**:

* Total number of gaps in the diagram (Integer).

### SwapCols()

**Lines:** 732-818

Function called by editSVG.html. Used to swap 2 specified columns in the diagram.

**Input**:

* Svg string from local storage.
* Column number 1.
* Column number2.

**Output**:

* Calls redrawOverlaps() function to put updated svg string into local storage.

### GetXLineEnd()

**Lines:** 819-833

Function used to get the x-Coordinate of the end of the lines section of the diagram.

**Input**:

* Array of lines.

**Output**:

* X axis value (Integer).

### GetXLineStart()

**Lines:** 834-848

Function used to get the x-coordinate of the start of the lines section of the diagram.

**Input**:

* Array of lines.

**Output**:

* X axis value (Integer).

### GetYAxisSpace()

**Lines:** 849-867

Function used to get y axis spacing for the diagram.

**Input**:

* Array of lines.
* Y axis start value for labels (Integer).
* Array of line numbers to be swapped.

**Output**:

* Y axis spacing between lines and labels (Integer).

### GetYStartLine()

**Lines:** 868-882

Function used to get the first y axis coordinate for lines.

**Input**:

* Array of lines.

**Output**:

* Y axis value (Integer).

### GetYstart()

**Lines:** 883-897

Function used to get the first y axis coordinate of labels

**Input**:

* Array of lines.

**Output**:

* Y axis value (Integer).

### ReDrawOverlaps()

**Lines:** 898-967

Function used to take the Array of rows and line spaces then rebuild them as the sections of the svg string. It will then rebuild the entire svg string.

**Input**:

* Array of rows.
* Arrays of column spaces.
* Array of svg sections.
* Array of Labels.
* X axis line start value (Integer).
* Array of x axis spacing (columns spacing).
* Y axis labels start value (Integer).
* Y axis line start value (Integer).
* Y axis spacing (Integer).
* Array of labels positions.
* Array of colors.
* Number of columns (Integer).

**Output**:

* Sets completed svg string to local storage.

### ChangeGuides()

**Lines:** 968-1011

Function used to change the x-axis spacing to reflect the changes in the diagram lines.

**Input**:

* Array of lines.
* Array of x axis spacing.
* Number of columns (Integers).

**Output**:

* New guides section of the diagram (sections[0]).

### Transpose()

**Lines:** 1012-1019

Function used to transpose the Array of rows to an Array of columns and vice versa. Treats the arrays as 2D matrices.

**Input**:

* Array of columns / rows.

**Output**:

* Array of rows / columns.

### GetLabels()

**Lines:** 1020-1034

Function used to get all labels used in the diagram.

**Input**:

* Array of lines.
* Array of line numbers which contain labels.

**Output**:

* Array of Labels.

### GetXPos()

**Lines:** 1035-1051

Function used to get X axis position for labels.

**Input**:

* Array of lines.
* Array of line numbers which contain labels.

**Output**:

* Array of x axis coordinates for labels.

### GetStrokes()

**Lines:** 1052-1066

Function used to get the colours used for lines.

**Input**:

* Array of lines.
* Array of line numbers which contain labels.

**Output**:

* Array of colours used per line.

### RowPop()

**Lines:** 1067-1114

Function used to convert the svg diagram into an array of Boolean values showing which column segment a line populates.

**Input**:

* Svg String.
* Number of columns (Integer).
* Array of x axis column spacing
* A axis start value for lines.

**Output**:

* Array of rows (Boolean values for each column).

### GetCols()

**Lines:** 1115-1120

Function used to get the number of columns in a diagram.

**Input**:

* X axis start value (Integer).
* X axis end value (Integer).
* Array of x axis spacing.

**Output**:

* Number of columns (Integer).

### SwapRows()

**Lines:** 1121-1207

Function called by editSVG.html. Used to swap 2 rows of the diagram.

**Input**:

* Svg string from local storage.
* Number of row 1.
* Number of row 2.

**Output**:

* Sets new svg string to local storage.

### GetLines()

**Lines:** 1208-1227

Function used to get line numbers for labels.

**Input**:

* Array of lines.

**Output**:

* Array of line numbers for labels.

### SeparateSVG()

**Lines:** 1228-1256

Function used to separate the svg into different sections.

Sections[0]: Start of the svg, contains opening tag and guides.

Sections[1]: Main section of the svg, contains labels and lines.

Sections[3]: End of the svg, contains the closing tag.

**Input**:

* Svg String passed in as variable.

**Output**:

* Array of sections.

### GetLineNo()

**Lines:** 1257-1267

Function used to get line numbers for labels.

**Input**:

* Array of lines.

**Output**:

* Array of line numbers for labels.

### GetHeight()

**Lines:** 1268-1282

Function used to get the height of the diagram.

**Input**:

* String input.

**Output**:

* Height of diagram as string (to be input into svg text).