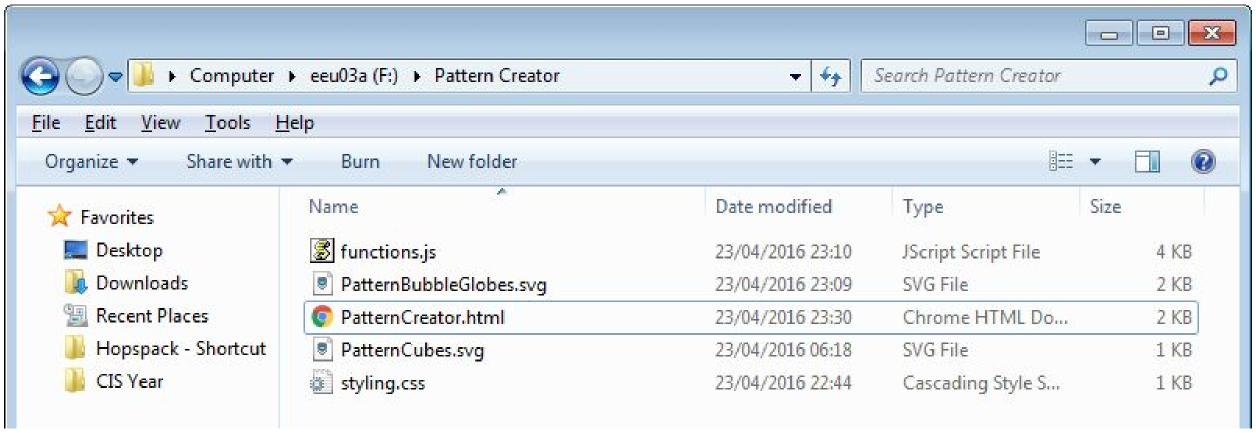
Pattern Creator Briefing Report

For this assignment I was tasked with creating an interactive pattern creator using SVG and Javascript using the document object model. Below are the files for the entire application.



Computer ▸ eeu03a (F:) ▸ Pattern Creator
Search Pattern Creator
File Edit View Tools Help
Organize Share with Burn New folder
Favorites
Desktop
Downloads
Recent Places
Hopspack - Shortcut
CIS Year

Name	Date modified	Type	Size
functions.js	23/04/2016 23:10	JScript Script File	4 KB
PatternBubbleGlobes.svg	23/04/2016 23:09	SVG File	2 KB
PatternCreator.html	23/04/2016 23:30	Chrome HTML Do...	2 KB
PatternCubes.svg	23/04/2016 06:18	SVG File	1 KB
styling.css	23/04/2016 22:44	Cascading Style S...	1 KB

The application was structured to give each pattern its own individual svg file, for ease of organisation. The HTML page gives a simple page structure, then the svg images were embedded in the page using <object> tags. I used this over <img> tags as the interactive aspect of the page would not function using img tags, as JS cannot alter an svg embedded in img tags. All of the javascript functions were stored in the functions.js so I only had to reference the file once.

Explanation of code

functions.js

All of the javascript functions in the file (apart from the enlarge function) manipulate the SVG directly. I chose not to use the JavaScript to draw SVG elements, but only alter the backbone SVG code I had written.

- rotatePattern(patternID) - Takes the ID of the svg object element as a parameter, then rotates the SVG 90 degrees clockwise. As the transform(rotate()) method takes a rotation angle, and the origin of the rotation as arguments, I had to first calculate the center of the SVG's bounding box then rotate 90 degrees. As the rotation method works with arguments between 0-360, when the rotation angle was at 270º I reset it back to 0 on the next iteration.
- invertColours() - For the bubbles-svg object, I selected specific elements from the SVG document to change the colour of. This was not fully inverting the colours of the image which could have been simply done, but I chose to do this to make the patterns better for the end user who would have been colouring it in.
- spaceOut() - For the cubes-svg object. This function made the lines in the pattern inside the boxes spaced further apart, to make colouring in easier. I did this by incrementing the height of the pattern by 1 each time to increase whitespace between lines.

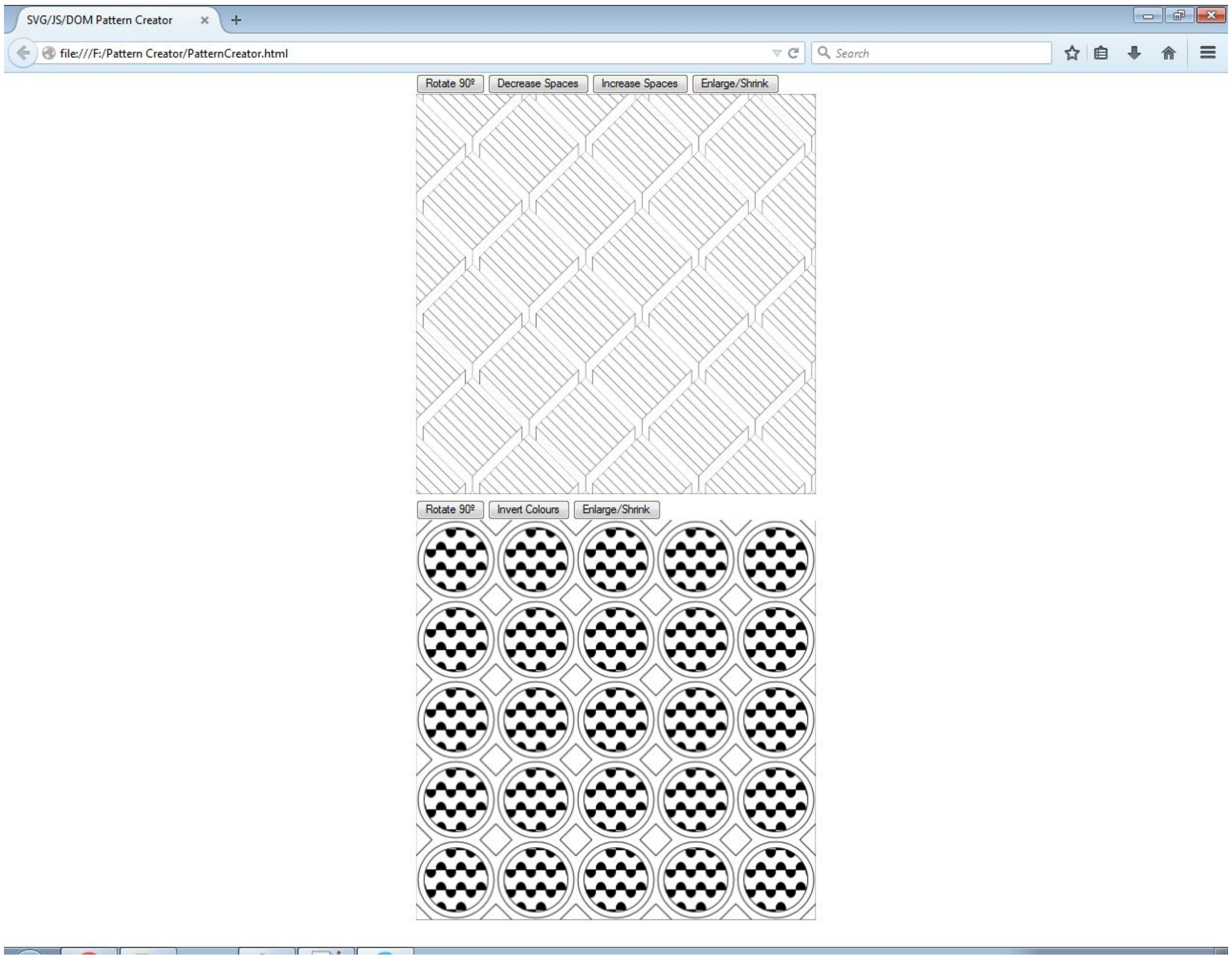
- spaceIn() - As for spaceOut, but the opposite way. Stops when the line pattern is 20 height as any smaller is not practical for colouring.
- enlargePattern() - The only pattern that does not manipulate the SVG. This function makes the button act as a toggle to enter/exit enlarged mode. It manipulates the CSS on the page to make the selected pattern and its buttons fit the screen height, and hides the other elements on the page. When the button is clicked when the pattern is enlarged, it reverses the effect.

patternBubbleGlobes.svg
Utilises 2 patterns inside a wrapper to create the pattern. Easier to explain as a tree.
PatternWholeWrapper (wrapper for entire pattern)
---circles (pattern consisting of two concentric circles, as well as lines in the corner to form squares between the circles.
------hemi (pattern that fills the innermost circle, comprised of a simple wave pattern, as well as a line that bisects the waveform)

patternCubes.svg
Utilises 2 patterns inside a wrapper to create the pattern. Again, tree.
PatternWholeWrapper (wrapper for entire pattern)
---boxes (pattern that is used to generate the 3D box effect, made with a single rectangle with 3 lines on 3 corners to simulate a 3D look.
------lines (a simple line pattern that fills the boxes. As the parent(boxes) is rotated 45º, the lines also appear rotated).

Styling.css
A few CSS properties for the pattern wrapper divs on the page.

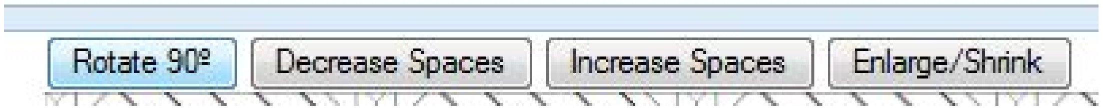
Screenshot of page on load



User Instructions

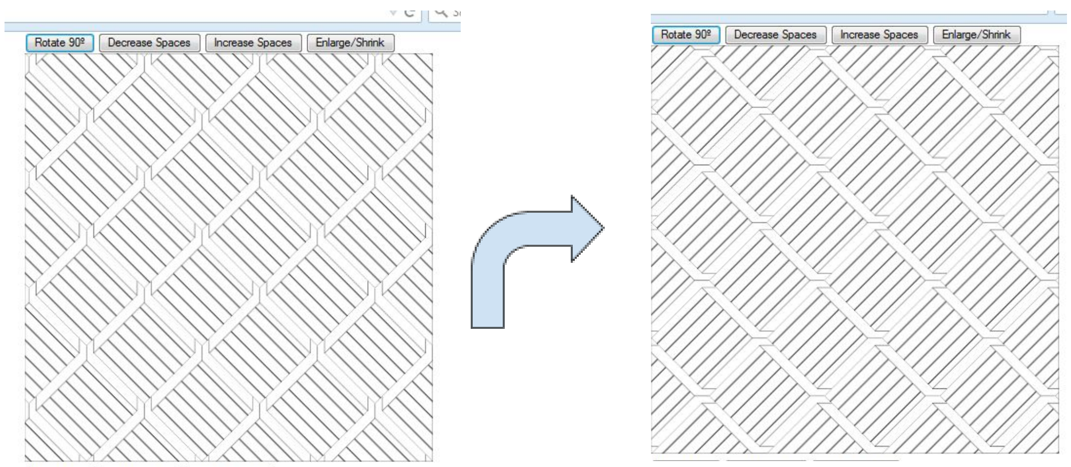
The user controls for the pattern creator are fairly intuitive, but I'll explain each button and show ____________________________________ what it does.

First Pattern



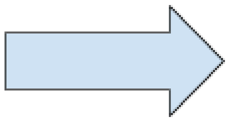
Rotate 90º

Does what it says on the tin, rotates the pattern by 90º.



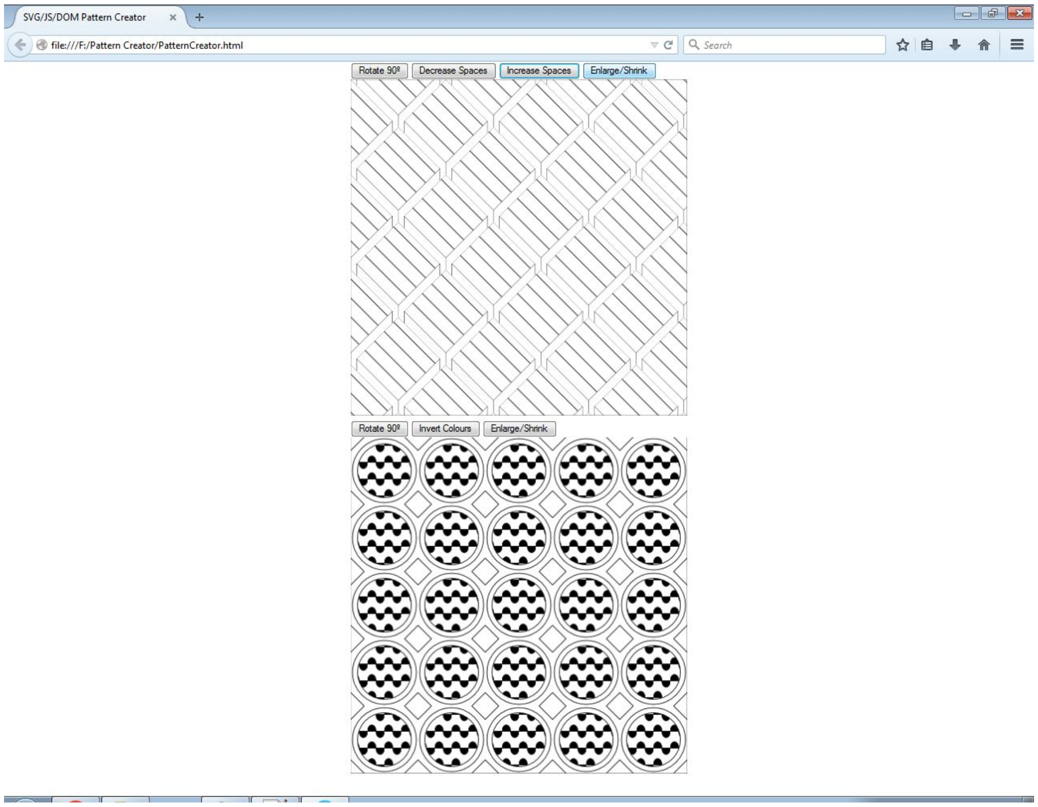
Decrease Spaces

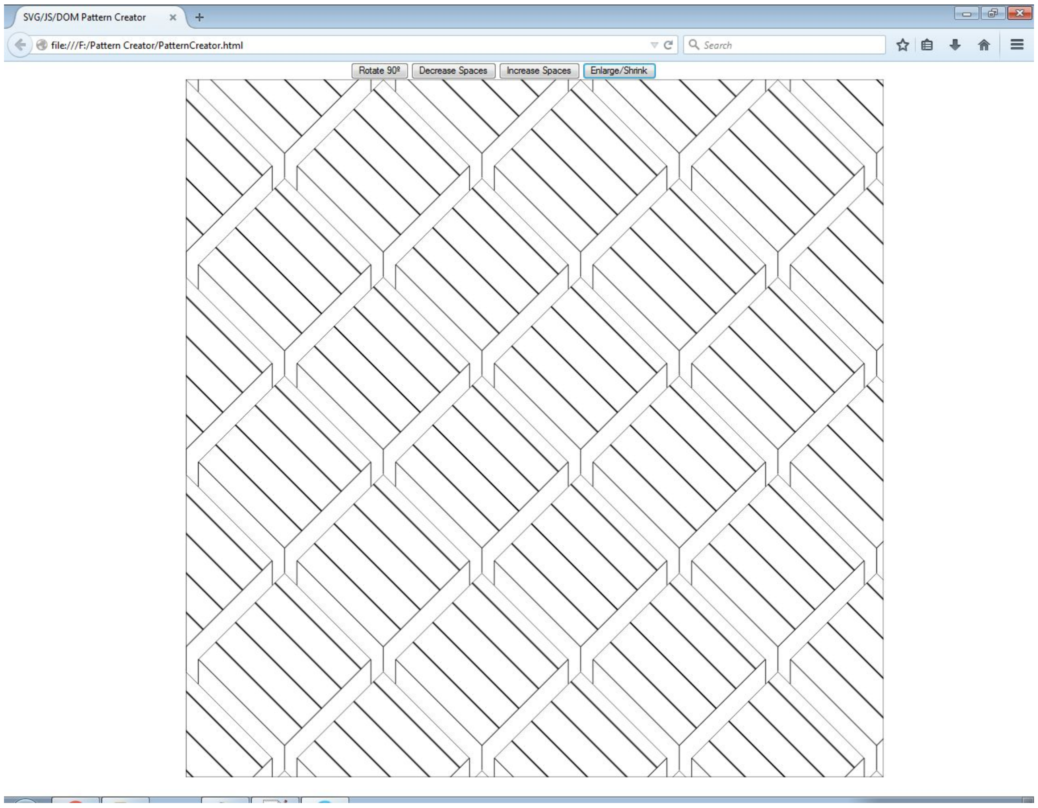
Decreases the spaces between lines in the boxes.



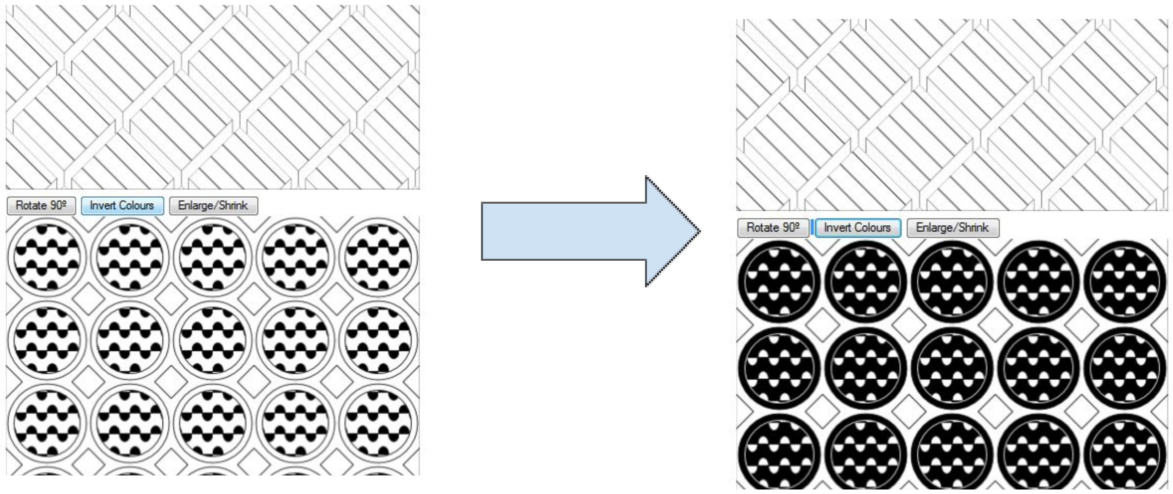
Increase Spaces just does the opposite.

Enlarge/Shrink
Fills the page with the pattern, hides the other. All controls maintain functionality when the pattern is enlarged.




SVG/JS/DOM Pattern Creator
file:///F:/Pattern Creator/PatternCreator.html
Search
Rotate 90º
Decrease Spaces
Increase Spaces
Enlarge/Shrink

Invert Colours
Inverts the colours on the bubbles pattern.



Critical Analysis
Ideally I would have liked to include more patterns to closer match the designs on my five design sheet, but I felt the time it took to code the two presented would've made it harder to include as much functionality if I had another two patterns as well, so I tried to match the designs as much as I could. The same also goes for the amount of manipulation controls in JavaScript, but I wanted the functions to work as described without bugs. Overall I feel I am now confident in coding in SVG and Javascript, and incorporating the two to work with the Document Object Model.

Websites Used
http://www.tutorialspoint.com/svg/
http://www.tutorialspoint.com/javascript/
http://svgtutorial.com/manipulating-svg-with-javascript/
http://www.petercollingridge.co.uk/data-visualisation/using-javascript-control-svg
https://stackoverflow.com/questions/21716097/manipulate-svg-file-using-javascript
http://tutorials.jenkov.com/svg
https://stackoverflow.com/questions/5581034/is-there-are-way-to-make-a-child-divs-width-wider-than-the-parent-div-using-css
https://www.codementor.io/tips/7514273358/manipulate-elements-in-svg-file-object-by-javascript-or-jquery
https://benfrain.com/selecting-svg-inside-tags-with-javascript/
https://stackoverflow.com/questions/6242976/javascript-hide-show-element