CS264 - Software Design

Assignment 02

Assignment Release Date:	10-10-2022
Submission Due Date:	21-10-2022
Feedback Due Date (estimated):	19-11-2022 (for assignments that make Due Date)
Support Laboratories	Labs 03-04 (Two Weeks)
Total Mark:	10%

This Assignment is worth 10% of the Software Design CA Component.

This is an open-book, graded assignment. You may use online resources for reference purposes only to help with the assignment. Please cite all references as comments in your submissions. You cannot directly reuse C# **solution code** from online sources. **You must not engage with another student, in person or electronically (phone, social media, etc.) to secure assistance with this assignment. If you do so you will receive an automatic fail (0%).** We will perform similarity checks on submitted assignments to check for collaborative efforts. A reasonable attempt at this assignment will gain you 10% of your continual assignment marks. It is possible to gain extra credit (up to a maximum of 5%) for this assignment.

Assignment 02 - Object Oriented Development with C#

You are required to design and develop an object oriented solution in C# for maintaining a data model for a basic vector graphics application. The model should include a means to represent a single two-dimensional (2D) canvas that maintains a list basic 2D shapes, e.g. rectangles, circles, etc. on a drawing canvas.

Note that you do not need to implement a graphical user interface for this assignment, i.e. you are not building a graphics app. This assignment is about designing and developing the classes and methods associated with a typical drawing app data model. However, the assignment does expect that you will convert the model into a graphic file viewable on a browser.

Assignment 02 - Requirements

You are required to implement the following functionality for data model for a basic vector graphics application consisting of a canvas and embedded shapes.

Shapes are drawn on (added to) a canvas which has a height and width. A canvas may contain many shapes and there can be only one drawing canvas. Shapes may overlap, and there may be an unlimited number of shapes present on the canvas. Shapes, where appropriate, have line styles, line thicknesses, line colours, fill colours, etc. Shapes have a stacking order which is referred to as a z-index. Your model must include z-index information about all shapes on the canvas.

Note that your solution should implement an object oriented design, and where appropriate implement a design incorporating design patterns and class hierarchies.

For your application you should provide the following core functionality:

1. Functionality for creating and managing the drawing canvas. Your application should be provide methods to export the canvas to a file in SVG (Support Vector Graphics) format.
2. Functionality for creating and managing shapes on the canvas, creating shapes, adding shapes to, deleting shapes from, and updating shapes on, the canvas. All shapes have an associated line style which may be changed from the default style. Closed shapes also have a fill style, i.e circles may be filled with blue colour, and have black dashed border lines, but lines don't have a fill colour (just a line colour).
3. Functionality to manage the z-index (drawing order) of the individual shapes on the canvas. Typically, z-index refers to the stack order of an element. An element with greater stack order is always in front of an element with lower stack order.
4. Functionality to set default line and fill styles for individual shape types, for example, default rectangles are grey with black 1pt solid line borders.

Assignment 02 - Basics Shapes that can be added to the canvas

Here are the attributes for the basic **Shapes** (elements) typically used in vector graphics applications, and should be used in your model:

- A **Rectangle** shape has four basic attributes that control the position and shape. **x** is the x position of the top left corner of the rectangle. **y** is the y position of the top left corner of the rectangle. **width** is the width of the rectangle. **height** is he height of the rectangle.
- A **Circle** shape has three basic attributes that control the position and size. **r** is the radius of the circle. **cx** is the x position of the centre of the circle. **cy** is the y position of the centre of the circle.
- An **ellipse** shape is a more general form of the circle element, where you can scale the **x** and **y** radius of the circle separately and has four basic attributes. **rx** is the x radius of the ellipse. **ry** is the y radius of the ellipse. **cx** is the x position of the centre of the ellipse. **cy** is the y position of the centre of the ellipse.
- A **line** shape takes the positions of two points as parameters and draws a straight line between them and has four basic attributes. **x1** is the x position of point 1. **y1** is the y position of point 1. **x2** is the x position of point 2. **y2** is the y position of point 2.
- A **polyline** shape is a group of connected straight lines and contains a single attribute which is a list of **points** specifying the individual lines. The end point (x and y coordinates) of one line becomes the starting point for the next line.
- A **polygon** shape is similar to a polyline shape, as it is also composed of straight line segments connecting a list of points to form a closed shape, and contains a single attribute which is a list of points specifying the individual lines. For polygon shapes, however, the path automatically connects the last point with the first.

- A **path** shape is the most general shape that can be used to draw shapes. The path shape may be used to draw all other shapes and curves. The path shape has a single attribute that contains a list of instructions describing the path.

Note that Shapes, where appropriate, also have line styles, line thicknesses, line colours, fill colours, etc.

Assignment 02 - Additional Requirements for Extra CA Credit (5%)

If you manage to develop and design and solution to this assignment before the deadline you may also wish to include (i) functionality for grouping and ungrouping selected shape elements on the canvas, (ii) applying styles and translations to shape groups, (iii) implementing canvas methods for translating, rotating, skewing and scaling basic shapes, and (iv) adding styled and formatted text to the canvas.

Successfully completing this extra functionality will result in awarding an extra 5% towards your CA Total.

Assignment 02 - References

These references may provide some assistance with understanding SVG generation.

https://www.w3.org/TR/SVG2/shapes.html

https://1stwebdesigner.com/create-simple-shapes-svg/

https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial

https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Basic_Shapes

https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Paths

IMPORTANT SUBMISSION DETAILS

Please indicate the Operating System (Linux/Windows/MacOS/Online) and IDE (e.g. VS Code) version used for testing (as a comment in your submitted code).

All work must be submitted via Moodle (see "Assignments" section for submission). Work submitted via other means will not be accepted unless you have prior arrangements with the Head Demonstrator (Mark McCormack). All work MUST be submitted by the due-date deadline. Late submissions will not be accepted.

If you are attempting the Extra Credit option you must upload your solution with the extra-credit code as a separate upload using the submission link provided in the CS264 Moodle space.

*The assignment submission is a zip file named "**assignment-02-xxxxxxxxx.zip**" (where "xxxxxxxxx" is your student id) containing solution files, e.g. named "**svgmodel.cs**", "**Program.cs**", etc. together with any other resources used in the assignment solution. Please ensure that all external files use relative directory referencing, rather than hard-coding the files' location.*