UML Report

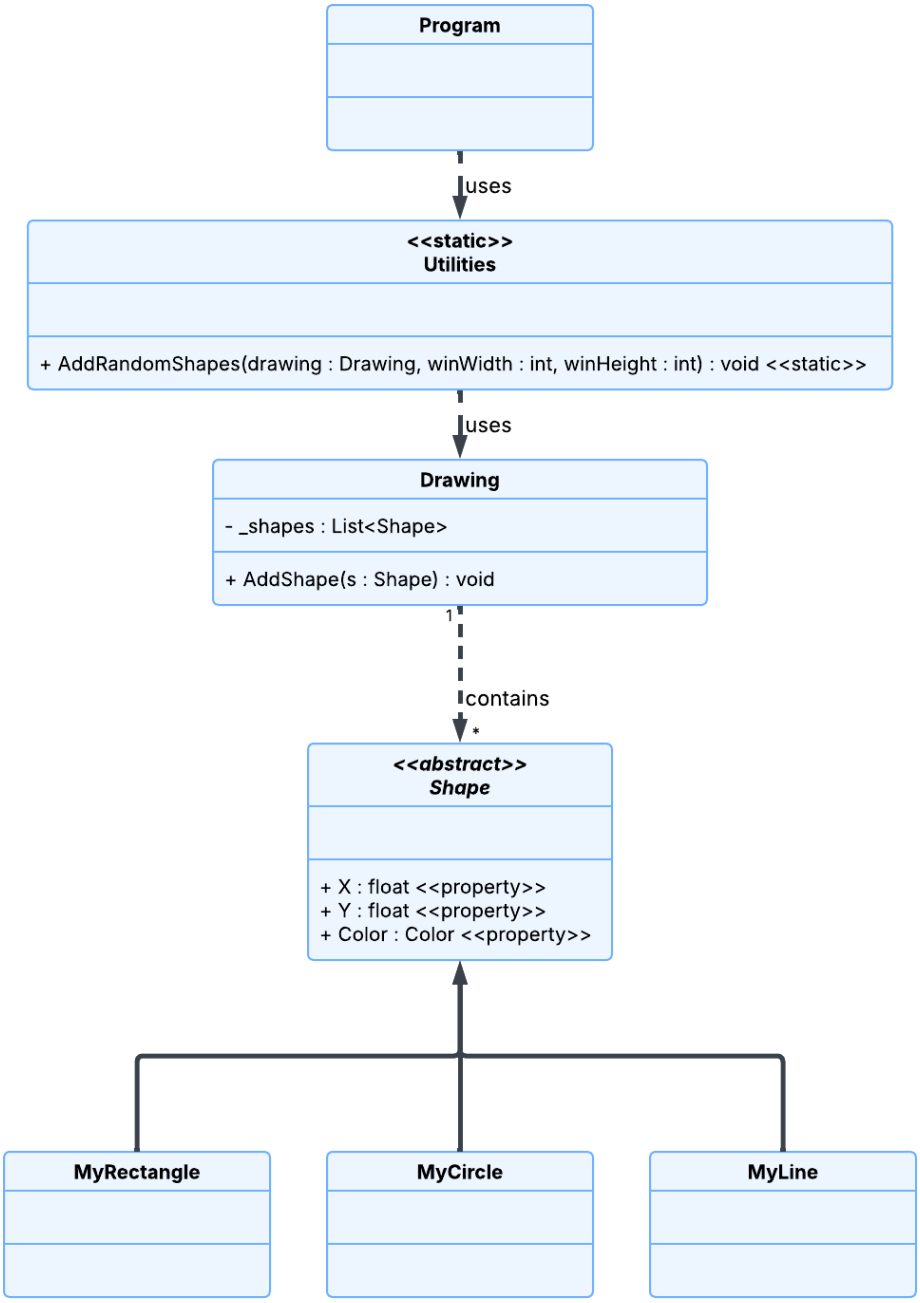
Cos20007

Dylan Rodwell: 105341089

105341089@student.swin.edu.au

Feature 2: Draw Random Shapes (D Key)

Class Diagram:



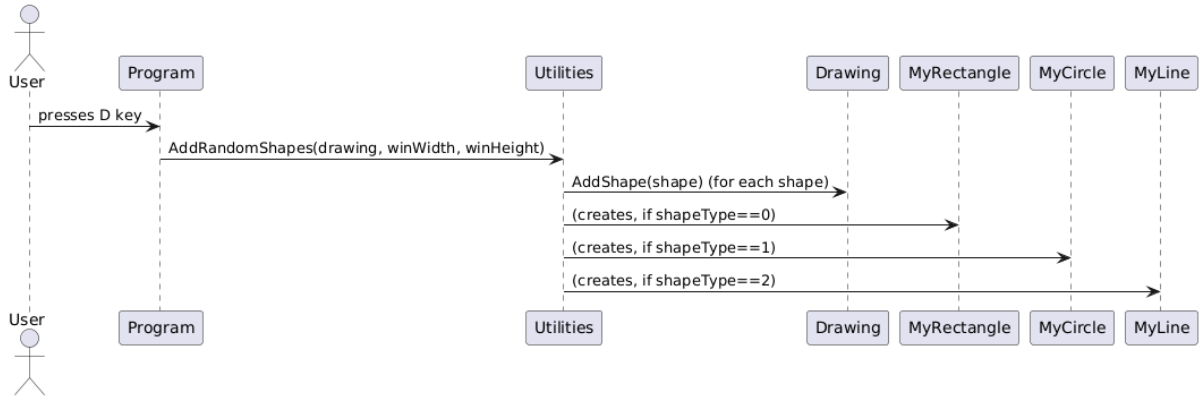
Description:

The class diagram above shows the static ‘Utilities’ class, which provides helper methods for scaling shapes, drawing a name using shapes, and adding random shapes to the canvas. This report will only focus on the feature that adds random shapes to the canvas. The ‘Program’ class uses these utilities to manipulate a ‘Drawing’ object, which aggregates multiple ‘Shape’ instances (‘MyRectangle’, ‘MyCircle’, ‘MyLine’).

Key Classes:

Class	Attributes/Methods	Relationships
Program	(main loop, event handling)	Uses ‘Utilities’
Utilities	‘+AddRandomShapes’	Uses ‘Drawing’
Drawing	‘-_shapes : List<Shape>’, ‘+AddShape(s : Shape)’	Contains many Shapes
Shape	‘+X : float’, ‘+Y : float’, ‘+Color : Color’	Abstract base for types of shapes
MyRectangle	‘+Width : float’, ‘+Height : float’	Inherits Shape
MyCircle	‘+Radius : float’	Inherits Shape
MyLine	‘+Length : float’	Inherits Shape

Sequence Diagram:



Description:

When the user presses the D key, the following occurs:

1. ‘Program’ detects the key press.
2. It calls ‘Utilities.AddRandomShapes’, passing the current ‘Drawing’ and window dimensions.
3. ‘Utilities’ creates several random shape objects and adds them to the ‘Drawing’ by calling ‘AddShape’.
4. Depending on the shape type, it creates a ‘MyRectangle’, ‘MyCircle’, or ‘MyLine’.

Analysis:

The design centralises stateless helper operations in the ‘Utilities’ static class, promoting code reuse and separation of concerns. The main program loop delegates all drawing and manipulation features to those helpers, keeping the code modular and maintainable.