

Comp2522 A#4

Purpose: To practice design and development with OOP principles in particular Inheritance and Polymorphism with class decoupling for future modifications.

Description: You are to design a simple drawing program. Your design is your own but must have the following:

User Interface (JavaFX) with buttons (buttons=checkboxes, dropboxes, pushbuttons, radio buttons) for shapes (see below), colour for drawing, selecting shapes, deleting shapes, moving shapes.

Shapes: Circle, Rectangle, Oval, Square, Polygon* (optional but worth 5 extra marks, must be selectable, movable). Note: Oval is a generalized Circle (they are related – hint – but also circle != Oval), same with Rectangle and Square. Polygon selection limited to convex polygons if you add this shape.

Colour: You can use a colour chooser or setup a dropbox of at least 10 colours

You must use each of the 3 types of design diagrams discussed in lecture: Sequence, Collaboration, and Class diagrams. **For the Sequence and Collaboration focus on User drawing shapes.** Create your diagrams following the style shown in the slides and using some form of drawing tools to create neat clear diagrams.

Help:

Your design needs to consider future possible changes.

The UI should be made of panels with functionality grouped. Each panel would be a special class containing the buttons for specific tasks. A parent panel would then hold all of these panels and would allow future features for the UI to be added by simply adding a new special panel class to the parent panel. In JavaFX the use of a Pane class type is used for each layout. By creating your own class to subclass a desired layout style of Pane (or if you prefer have it use the Pane class “has-a”) you can define your panels.

Shapes – shapes should track their points that they need to be drawn and their colour. For basic shapes this is two points (start and end). Shapes are drawn by clicking and dragging or for Polygon multiple clicks. **Rubber banding is simulated by using a new Canvas for each new shape, clearing the canvas during the drawing process before redrawing it** (Javafx uses a delay drawing technique to reduce flicker). This prevents flicker and gives the illusion of rubber banding. Think about how you can handle new shapes being added that are non-standard e.g. Polygon – contains more than 2 points. Is your shape class able to deal with possible new types of shapes? The other classes should not know (or need to) what type of shape is being drawn this includes the event handlers. Since the user may swipe up rather than down when drawing shapes, you need to determine while drawing which is the top left corner and which is the bottom right corner based on the initial click point and the current mouse position e.g. User

clicks at 100,100 then moves to 120, 80. You must create two different points from this to be 100,80 and 120, 100 (which would create the same rectangle but use the correct left,top,right,bottom points as required by Java when drawing rectangles and ovals).

You need to store each shape and associated canvas.

To select a shape, the user enters select mode (pressing the select button/checkbox/whatever), then clicks the mouse on the desired shape. Since shapes may lay on top of each other, order of drawing matters. To help with this, use a stack type to track shapes. Since the shapes are stacked in the order they were drawn, searching the stack from top down will allow testing of each shape in correct order. To determine if the shape was selected, using the mouse position at the time of the click, check if this position is within the bounds of the shape.

Selected shape can be moved or deleted (dragged or delete key pressed). To drag, clear that canvas and draw where the mouse is using the closest point of the mouse as reference.

Can your design handle future possible features? How much coupling between classes is there? Marks will be awarded for how flexible your design is. Code must match your design that you hand in.

Marking

Function	Mark
Design	30
Shapes	20
Rubber banding, all direction dragging	20
Selecting	5
Moving	5
Deleting	5
Colour selection	5
Documentation and commenting	10

DUE: April 7 at midnight.



