Group 21: Paint

● ● ●

Jonathan Vu, Robert Allen

Demo: https://www.youtube.com/watch?v=z-x1WGTwCzQ

Design Pattern: Singleton

Paint
-paint: Paint
+getInstance(): Paint
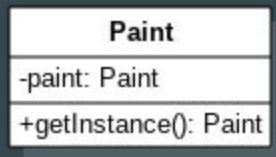
Design Pattern: Observer
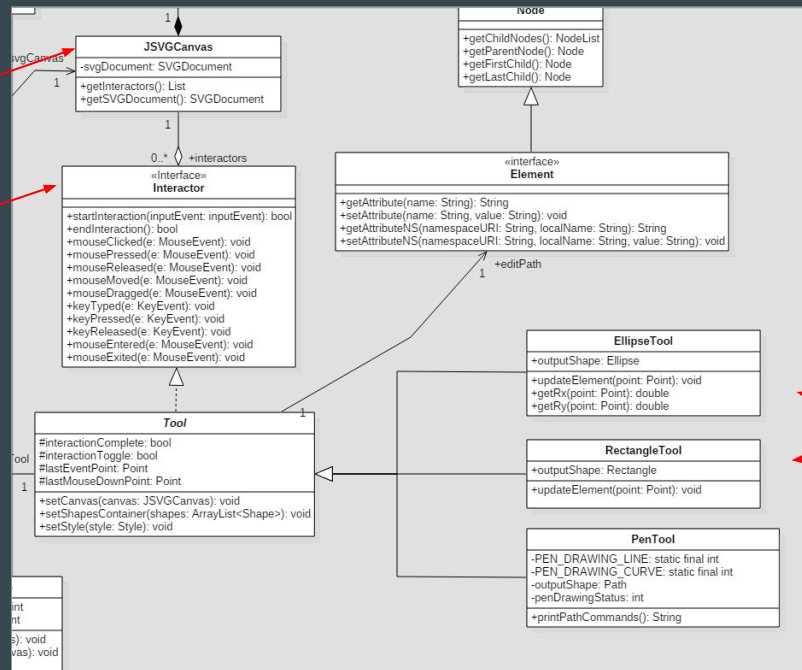
Design Pattern: Composite

Shapes

The Canvas only needs to deal with Shape. More shapes can be added without changing Canvas

Tools
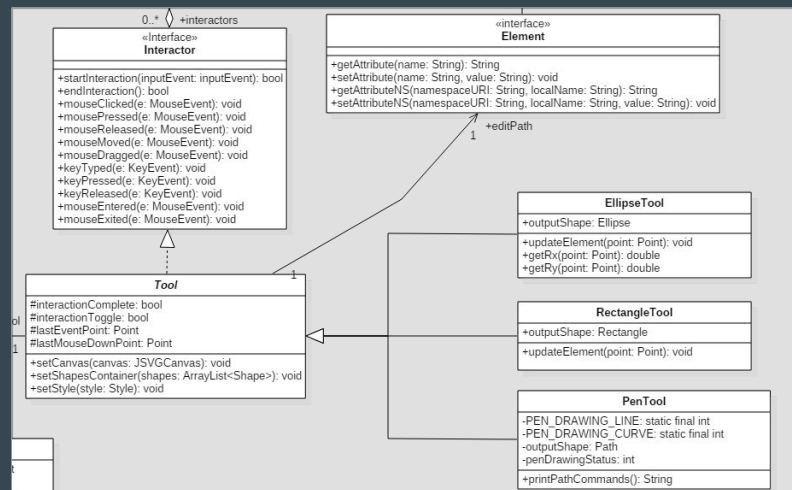
Subjects in observer pattern only need to deal with Tool. Tool types can be added without changing subjects.

Final Class Diagram

Paint
-window: JFrame
-paint: Paint
-Paint()
+getInstance(): Paint
-createMenuBar(): void
-createGUI(): void
-createToolOptionsBar(): void
-createToolSideBar(): void

JButton
+command: AbstractAction

«interface»
SVGDocument
+getTitle(): String
+getReferrer(): String
+getDomain(): String
+getURL(): String
+getRootElement(): Element

«interface»
Document
+getDocumentElement(): Element
+createElement(tagName: String): Element
+createElementNS(namespaceURI: String, tagName: String): Element

AbstractAction
+name: String
+actionPerformed(): void

SaveDialog
-savePath: JTextField
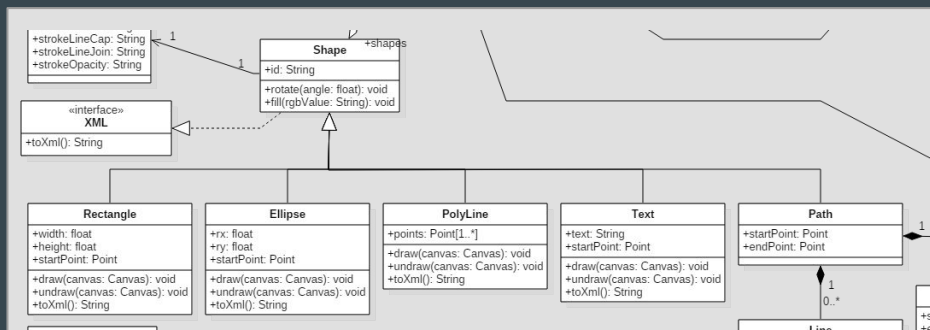-saveButton: JButton
-cancelButton: JButton
-save(): void

LoadDialog
-loadPath: JTextField
-loadButton: JButton
-cancelButton: JButton
-load(): void

ResizeDialog
-newWidth: JTextField
-newHeight: JTextField
-okButton: JButton
-cancelButton: JButton
-resize(width: int, height: int): void

JSVGCanvas
-svgDocument: SVGDocument
+getInteractors(): List
+getSVGDocument(): SVGDocument

«interface»
Node
+getChildNodes(): NodeList
+getParentNode(): Node
+getFirstChild(): Node
+getLastChild(): Node

SVGFile
-file: File
+save(): void
+open(): void

Canvas
-width: int
-height: int
+draw(): void
+rotate(angle: float): void
+resize(newWidth: int, newHeight: int): void
+crop(x1: int, y1: int, x2: int, y2: int): void
+setTool(tool: Tool): void
-getSvg(): String

RotateDialog
-cw: bool
-angle: int
-clockWise: JRadioButton
-counterClockWise: JRadioButton
-okButton: JButton
-cancelButton: JButton
-rotate(cw: bool, angle: float): void

«Interface»
Interactor
+startInteraction(inputEvent: inputEvent): bool
+endInteraction(): bool
+mouseClicked(e: MouseEvent): void
+mousePressed(e: MouseEvent): void
+mouseReleased(e: MouseEvent): void
+mouseMoved(e: MouseEvent): void
+mouseDragged(e: MouseEvent): void
+keyTyped(e: KeyEvent): void
+keyPressed(e: KeyEvent): void
+keyReleased(e: KeyEvent): void
+mouseEntered(e: MouseEvent): void
+mouseExited(e: MouseEvent): void

«interface»
Element
+getAttribute(name: String): String
+setAttribute(name: String, value: String): void
+getAttributeNS(namespaceURI: String, localName: String): String
+setAttributeNS(namespaceURI: String, localName: String, value: String): void

Transform
-matrix: String
-translate: String
-scale: String
-rotate: String
-skewX: String
-skewY: String

Style
-fill: String
-fillRule: String
-stroke: String
-strokeWidth: String
-strokeLineCap: String
-strokeLineJoin: String
-strokeOpacity: String

EllipseTool
+outputShape: Ellipse
+updateElement(point: Point): void
+getRx(point: Point): double
+getRy(point: Point): double

Shape
+id: long
+rotate(angle: int): void
+fill(rgbValue: String): void
+toXml(): String
~docToString(doc: Document): String
-setGlobalAttributes(elem: Element): void

«interface»
XML
+toXml(): String

Tool
#interactionComplete: bool
#interactionToggle: bool
#lastEventPoint: Point
#lastMouseDownPoint: Point
+setCanvas(canvas: JSVGCanvas): void
+setShapesContainer(shapes: ArrayList<Shape>): void
+setStyle(style: Style): void

RectangleTool
+outputShape: Rectangle
+updateElement(point: Point): void

Rectangle
+width: float
+height: float
+startPoint: Point
+draw(canvas: Canvas): void
+undraw(canvas: Canvas): void
+toXml(): String

Ellipse
+rx: float
+ry: float
+startPoint: Point
+draw(canvas: Canvas): void
+undraw(canvas: Canvas): void
+toXml(): String

PolyLine
+points: Point[1..*]
+draw(canvas: Canvas): void
+undraw(canvas: Canvas): void
+toXml(): String

Text
+text: String
+startPoint: Point
+draw(canvas: Canvas): void
+undraw(canvas: Canvas): void
+toXml(): String

Path
+startPoint: Point
+endPoint: Point

PenTool
-PEN_DRAWING_LINE: static final int
-PEN_DRAWING_CURVE: static final int
-outputShape: Path
-penDrawingStatus: int
+printPathCommands(): String

Point
+x: float
+y: float

BezierCurve
+startControlPoint: Point
+endControlPoint: Point
+draw(canvas: Canvas): void
+undraw(canvas: Canvas): void
+toXml(): String

Line
+draw(canvas: Canvas): void
+undraw(canvas: Canvas): void
+toXml(): String