

제목	Delete기능과 도형이 추가된 그림판	과제번호	4
학과	독일어과	학번	201602173
이름	윤영택	제출일	20200528

1. 문제 정의

GrimPan3FXML을 수정하여 Delete 기능과 도형(별, 나선형, 하트 중 1개 이상)을 추가한다.

그 후, 멋진 그림을 만들어 save한다.

2-1. 문제 해결 방법

큰 문제를 두 가지의 작은 문제로 나누어 해결하고, 나눈 문제를 더 작은 문제로 분할하여 해결한다.

1) delete 기능을 어떻게 구현할 것인가?

delete 기능을 구현하기 위해서 두 가지 event를 다뤄야 한다. 첫 번째는 그림판 메뉴 중에서 **delete가 선택되었을 때** 발생하는 event이다. 이때, 에디터의 상태가 delete모드가 되도록 한다. 이를 위해 handleMenuDelete()메소드 내부와 Utils.java에 약간의 코드를 추가한다(EDIT_DELETE). 두 번째로 다뤄야 할 event는 삭제하기 위해 **마우스를 조작했을 때** 발생하는 event이다. 이 문제는 move 기능과 관련된 코드를 참고하여 해결한다. 마우스가 클릭되면 커서가 놓인 곳에 위치한 도형이 선택되게끔 한다. 그 후, 마우스를 손에서 떼면 선택된 도형이 삭제되게 한다. 이를 위해 도형을 선택하는 메소드와 삭제하는 메소드를 각각 구현한다.

2) 어떤 도형을 어떻게 추가할 것인가?

기존에 있던 Regular를 응용하여 별을 추가한다. 기본적인 작동 원리는 Regular와 거의 흡사하며, ShapeFactory 내부에서 그림을 그리는 함수의 코드만 수정한다. 다만, Regular는 사용자로부터 입력을 받아 도형을 생성하지만, 별의 모양을 5각형으로 고정하고자 handleMenuStar()에서 입력을 받는 부분은 생략한다.

2-2. 설계 결과(프로그램 구성도, 필요 함수, 클래스 설명)

1) 프로그램 구성도

- 이미지 크기의 문제로 다음 페이지에

```

<<Java Class>>
GrimPan3FXMLMain
hufs.ces.grimpan

GrimPan3FXMLMain()
start(Stage):void
main(String[]):void

```

```

<<Java Class>>
Utils
hufs.ces.grimpan

DEFAULT_DIR: String
MINPOLYDIST: int
SHAPE_REGULAR: int
SHAPE_OVAL: int
SHAPE_POLYGON: int
SHAPE_LINE: int
SHAPE_PENCIL: int
SHAPE_STAR: int
EDIT_MOVE: int
EDIT_DELETE: int
SHAPE_NAME: String[]

Utils()
getExtension(String):String

```

```

<<Java Class>>
GrimPanFXController
hufs.ces.grimpan

parentStage: Stage
STAR: int
widthProp: DoubleProperty
heightProp: DoubleProperty
fcolorPicker: ColorPicker
scolorPicker: ColorPicker
lbtSize: Label
lbtCount: Label
root: AnchorPane
drawPane: Pane
menuNew: MenuItem
menuOpen: MenuItem
menuSave: MenuItem
menuSaveAs: MenuItem
menuExit: MenuItem
menuLine: RadioMenuItem
menuPencil: RadioMenuItem
menuPolygon: RadioMenuItem
menuRegular: RadioMenuItem
menuOval: RadioMenuItem
menuStar: RadioMenuItem
menuMove: MenuItem
menuDelete: MenuItem
menuStrokeWidth: MenuItem
menuStrokeColor: MenuItem
menuFillColor: MenuItem
menuCheckStroke: CheckMenuItem
menuCheckFill: CheckMenuItem
menuAbout: MenuItem

GrimPanFXController(Stage)
initDrawPane():void
clearDrawPane():void
redrawDrawPane():void
handleMenuAbout(ActionEvent):void
handleMenuCheckFill(ActionEvent):void
handleMenuCheckStroke(ActionEvent):void
handleMenuDelete(ActionEvent):void
handleMenuExit(ActionEvent):void
handleMenuFillColor(ActionEvent):void
handleMenuLine(ActionEvent):void
handleMenuMove(ActionEvent):void
handleMenuNew(ActionEvent):void
handleMenuOpen(ActionEvent):void
handleMenuOval(ActionEvent):void
handleMenuPencil(ActionEvent):void
handleMenuPolygon(ActionEvent):void
handleMenuRegular(ActionEvent):void
handleMenuSave(ActionEvent):void
handleMenuStar(ActionEvent):void
handleMenuStrokeColor(ActionEvent):void
handleMenuStrokeWidth(ActionEvent):void
handleMenuSaveAs(ActionEvent):void
handleMouseEntered(MouseEvent):void
handleMouseExited(MouseEvent):void
handleMouseDragged(MouseEvent):void
handleMousePressed(MouseEvent):void
handleMouseReleased(MouseEvent):void
openAction():void
readShapeFromSVGSaveFile(File):void
saveAction():void
saveAsAction():void
saveGrimPanSVGShapes(File):void
moveShapeByMouse():void
endShapeMove():void
deleteShape():void

```

```

<<Java Class>>
SVGPath2PathHandler
hufs.ces.grimpan.svg

absX: float
absY: float
savX2: float
savY2: float
path: Path

SVGPath2PathHandler(Path)
startPath():void
moveToRel(float, float):void
moveToAbs(float, float):void
endPath():void
closePath():void
lineToRel(float, float):void
lineToAbs(float, float):void
lineToHorizontalRel(float):void
lineToHorizontalAbs(float):void
lineToVerticalRel(float):void
lineToVerticalAbs(float):void
curvetoCubicRel(float, float, float, float, float, float):void
curvetoCubicAbs(float, float, float, float, float, float):void
curvetoCubicSmoothRel(float, float, float, float):void
curvetoCubicSmoothAbs(float, float, float, float):void
curvetoQuadraticRel(float, float, float, float):void
curvetoQuadraticAbs(float, float, float, float):void
curvetoQuadraticSmoothRel(float, float):void
curvetoQuadraticSmoothAbs(float, float):void
arcRel(float, float, float, boolean, boolean, float, float):void
arcAbs(float, float, float, boolean, boolean, float, float):void
getPath():Path

```

```

<<Java Class>>
SVGUtils
hufs.ces.grimpan.svg

SVGUtils()
convertShapeToSVGPath(SVGGrimShape):SVGPath
convertSVGPathToPath(SVGPath):Path
getElementFromSVGPath(SVGPath):String
getPath2DFromSwingShape(Shape):Path2D
getPathFromSwingShape(Shape):String
getSVGStyleAttribute(Shape):String
getSVGStyleString(Shape):String
paintShapeWithSVGStyle(Shape, String):void
getColorFromStyleString(String):Color
getColorFromRGBString(String):Color
getStrokeWidth(String):double
rgbColor(Color):String

```

```

<<Java Class>>
SVGPath2PathParser
hufs.ces.grimpan.svg

path: Path

SVGPath2PathParser(String)
getPath():Path

```

```

<<Java Class>>
GrimPanModel
hufs.ces.grimpan

editState: int
shapeStrokeWidth: float
shapeStrokeColor: Color
shapeStroke: boolean
shapeFill: boolean
shapeFillColor: Color
startMousePosition: Point2D
currMousePosition: Point2D
prevMousePosition: Point2D
mouseInside: boolean
polygonPoints: ArrayList<Point2D>
selectedShape: int
nPolygon: int
saveFile: File
selectedShapeIndex: int

GrimPanModel()
getEditState():int
setEditState(int):void
getStartMousePosition():Point2D
setStartMousePosition(Point2D):void
getPrevMousePosition():Point2D
setPrevMousePosition(Point2D):void
getCurrMousePosition():Point2D
setCurrMousePosition(Point2D):void
getSaveFile():File
setSaveFile(File):void
getNPolygon():int
setNPolygon(int):void
getSelectedShape():void
getUnDeletedShape():void
getShapeStrokeColor():Color
setShapeStrokeColor(Color):void
isShapeFill():boolean
setShapeFill(boolean):void
getShapeFillColor():Color
setShapeFillColor(Color):void
getShapeStrokeWidth():float
setShapeStrokeWidth(float):void
isShapeStroke():boolean
setShapeStroke(boolean):void
getSelectedShapeIndex():int
setSelectedShapeIndex(int):void
isMouseInside():boolean
setMouseInside(boolean):void

```

```

<<Java Class>>
ShapeFactory
hufs.ces.grimpan

ShapeFactory(GrimPanModel)
getInstance(GrimPanModel):ShapeFactory
createPaintedShape(Shape):Shape
createPaintedEllipse():Ellipse
createPaintedLine():Line
createPaintedPath():Path
createMousePointedLine():Shape
createPolygonFromClickedPoints():Shape
createPolylineFromClickedPoints():Shape
createMousePointedEllipse():Shape
createStarPolygon(int):Shape
createRegularPolygon(int):Shape
translateShape(Shape, double, double):void

```

```

<<Java Class>>
SaxSVGPathParseHandler
hufs.ces.grimpan.svg

SaxSVGPathParseHandler()
startElement(String, String, String, Attributes):void
getPathList():ObservableList<SVGGrimShape>

```

```

<<Java Class>>
SVGGrimLine
hufs.ces.grimpan.svg

SVGGrimLine(Line)
getSVGShapeString():String
getPath2DShape():Path2D
main(String[]):void

```

```

<<Java Class>>
SVGGrimEllipse
hufs.ces.grimpan.svg

SVGGrimEllipse(Ellipse)
getSVGShapeString():String
getPath2DShape():Path2D
main(String[]):void

```

```

<<Java Class>>
SVGGrimPath
hufs.ces.grimpan.svg

SVGGrimPath(Path)
getSVGShapeString():String
getPath2DShape():Path2D
main(String[]):void

```

```

<<Java Class>>
SVGGrimPolyline
hufs.ces.grimpan.svg

SVGGrimPolyline(Polyline)
getSVGShapeString():String
getPath2DShape():Path2D
main(String[]):void

```

```

<<Java Class>>
SVGGrimPolygon
hufs.ces.grimpan.svg

SVGGrimPolygon(Polygon)
getSVGShapeString():String
getPath2DShape():Path2D
main(String[]):void

```

```

<<Java Class>>
SVGGrimStar
hufs.ces.grimpan.svg

SVGGrimStar(Path)
getSVGShapeString():String
getPath2DShape():Path2D
main(String[]):void

```

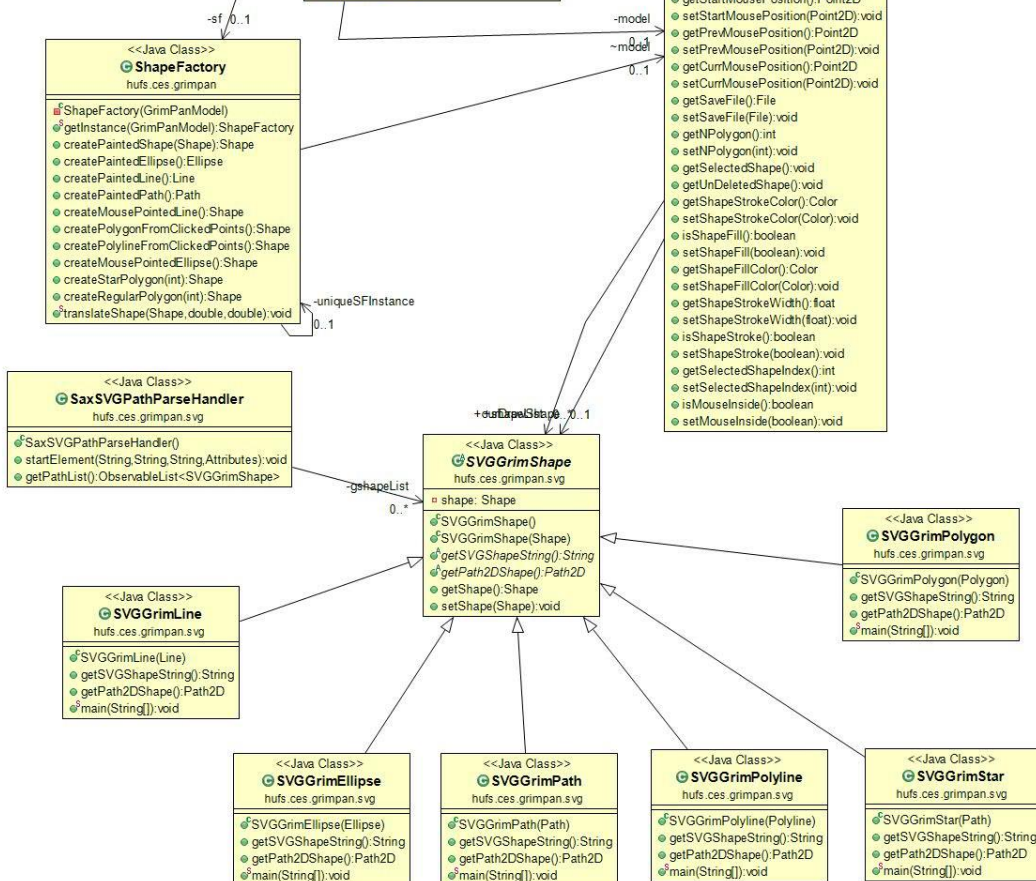
```

<<Java Class>>
SVGGrimShape
hufs.ces.grimpan.svg

shape: Shape

SVGGrimShape()
SVGGrimShape(Shape)
getSVGShapeString():String
getPath2DShape():Path2D
getShape():Shape
setShape(Shape):void

```



2) 필요함수, 클래스 설명

1) public void getUnDeletedShape()

기존에 있던 GrimPanModel.getSelectedShape()을 응용했다. shapeList내에서 선택된 도형의 index를 객체 변수에 저장한다.

2) private void deleteShape()

저장된 index를 바탕으로 해당 도형을 drawPane과 shapeList에서 제거한다.

3) public Shape createStarPolygon(int nvertex)

Shape Factory 내부에서 별모양을 그리는 함수이다. 별 모양은 정다각형과는 다르게 회전하는 각도가 일정하지 않으므로 Rotate객체를 각각 2개 만들고 polyPoints배열의 사이즈를 16으로 고정해서 짝수 index와 홀수 index가 각각 다르게 회전하도록 한다(조건문 사용). 이 경우 별 외부에 정오각형의 테두리가 생기는데, 불필요한 선들은 빼고 나머지 선들을 polygonPath에 추가해서 도형을 만든다(3.1에서 추가설명).

3. 결론 및 소감

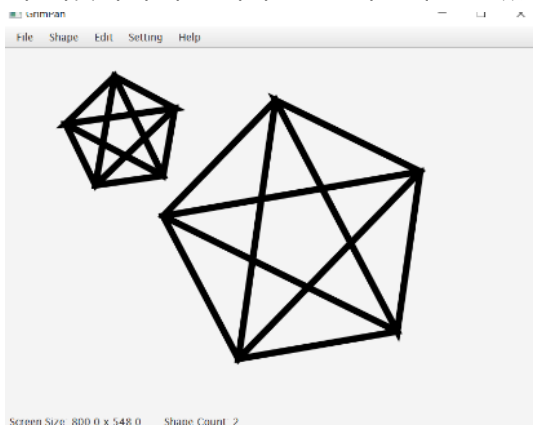
3.1 과제 해결 중 생긴 문제와 해결 과정

문제 1: 도형을 삭제해도 다른 곳을 클릭하면 삭제된 도형이 다시 생겼다.

-> 이 문제를 해결하기 위해 밤을 샜으나 정작 해결은 아주 간단했다. drawPane에서만 도형을 제거했기 때문에 발생한 문제였다. shapeList에서도 제거한 도형을 제거하니 완벽하게 해결했다.

문제 2: 별모양을 만들어지지 않았다.

-> 처음엔 SVGPath객체로 별을 만들고자 SVG Path Builder를 이용해 열심히 별을 그렸지만, 프로그램이 워낙 복잡하고 클래스들이 서로 연관되어 있기 때문에 독자적으로 SVGPath객체를 이용해 별을 그리는 건 포기했다. 결국 차선책으로 regular polygon을 이용해서 코드를 짰다. 다른 건 괜찮았지만 별모양을 그리기 위해 ShapeFactory에서 함수를 만드는 것은 쉽지 않았다. 예전에 파이썬에 내장된 터틀 그래픽으로 별모양을 그렸던 것을 떠올려 비슷하게 구현해서 별을 마침내 만들었다. 그런데, 큰 문제가 발생했다. 별 외곽에 정오각형이 있는 것이었다.



(사진 참조)

그림을 다시 그리기보단 "불필요한 선을 제거해서 polygonPath에 추가하면 되지 않을까?"라는 아이디어를 떠올렸다. 몇 번의 수정 후 마침내 별 모양을 완성했다.

3.2 배운 점 & 소감

지금까지는 EventHandle이 많지 않았는데, 이번 과제는 EventHandle에 관련된 부분이 정말 많았다. 특히 delete 기능을 구현하기 위해 마우스에 관련된 event를 처리하는 과정에서 무수한 에러가 발생했다. 결국 시간이 답이었다. 밤을 새고 계속해서 구글링을 하고 API를 뒤져보았다. 어떻게 해서 delete를 구현하고 보니 도형을 만드는 것이 남아있었다. 정말 여기서 많은 생각을 했다. 제한된 시간 때문에 기능적인 부분에서 타협도 고민했지만(예를 들어, 크기가 줄거나 늘어나지 않는 고정된 사이즈의 별을 만드는), 오기가 허락하지 않았다. 결국 시간을 갈아 넣은 끝에 프로그램을 완성했다. 뿌듯하기도 하지만 다음 과제가 두렵기도 하다. 결국은 시간과의 싸움인 것 같다.

3.3 그린 그림

