Hunter Simon
Period 4

Method Madness Essay

My project “Simon_4_MethodMadness.java” draws an amazing image using a total of eight methods. For example, my code prints a beautiful black and white penguin, with a narrow yellow beak and a gray outline around the eyes and chest. Personally, I really enjoy my project, I don't know why, but every time I look into the eyes of the penguin, it makes me smile. My program runs from methods know as *private void drawBody(gc);*. The public method calls the private method *private voidrawBody(GraphicsContext gc) {*.

```
@Override
public void start(Stage primaryStage) {
    primaryStage.setTitle("Drawing Operations Test");
    Group root = new Group();
    Canvas canvas = new Canvas(500, 500);
    GraphicsContext gc = canvas.getGraphicsContext2D();
    drawBody(gc);
    drawFeet(gc);
    drawArms(gc);
    drawBodyOutline(gc);
    drawChestOutline(gc);
    drawEyeWhites(gc);
    drawPupils(gc);
    drawBeak(gc);
    root.getChildren().add(canvas);
    primaryStage.setScene(new Scene(root));
    primaryStage.show();
}

private void drawBody(GraphicsContext gc) {
    gc.setFill(BLACK);
    gc.fillOval(150, 50, 200, 400);
```

Hunter Simon
Period 4

I used these methods.

```
private void drawBody(GraphicsContext gc) {
    gc.setFill(BLACK);
    gc.fillOval(150, 50, 200, 400);

}

    private void drawBodyOutline(GraphicsContext gc) {

    gc.setFill(GRAY);
    gc.fillOval(183, 97.5, 55, 55);
    gc.fillOval(172.5, 238, 155, 205);

}

    private void drawChestOutline(GraphicsContext gc) {

    gc.setFill(GRAY);
    gc.fillOval(257.5, 97.5, 55, 55);

}

    private void drawEyeWhites(GraphicsContext gc) {

    gc.setFill(WHITE);
    gc.fillOval(175, 240, 150, 200);
    gc.fillOval(185, 100, 50, 50);
    gc.setFill(WHITE);
    gc.fillOval(260, 100, 50, 50);

}

    private void drawPupils(GraphicsContext gc) {

    gc.setFill(BLACK);
    gc.fillOval(204, 117, 10, 10);
    gc.fillOval(280, 117, 10, 10);
private void drawBeak(GraphicsContext gc) {

gc.setFill(YELLOW);
int shiftx = 90;
int shifty = -140;
gc.fillPolygon(new double []{130 + shiftx, 160 + shiftx, 190 + shiftx},
        new double[]{300 + shifty, 340 + shifty, 300 + shifty},3);
gc.setFill(BLACK);
gc.fillRect(249, 160, 2, 40);
gc.fillOval(235, 165, 10, 10);
gc.fillOval(254, 165, 10, 10);



private void drawArms(GraphicsContext gc) {
gc.setFill(BLACK);
gc.fillOval(100, 200, 200, 30);
gc.fillOval(200, 200, 200, 30);




private void drawFeet(GraphicsContext gc) {

gc.setFill(BLACK);
gc.fillRect(220, 440, 3, 30);
gc.fillRect(277, 440, 3, 30);
```

The values I used for each method, for example gc.fillRect(100, 200, 200, 30); are coordinates and the length and height of the rectangle on the canvas. JavaFX automatically gives you a template,

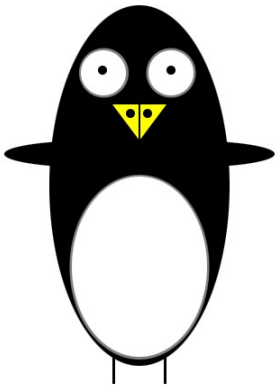
and that would be gc.fillRect(x, y, w, h);. Methods that are seen above can be used and be seen as gc.setFill(BLACK); passes a color instead of numbers. This method sets the color for all of the fills of a shape after that method. The shapes that create the pictures are drawn on the canvas. The canvas is a plate that you create using code and adding dimensions. My dimensions were 500x500. This creates a canvas in the shapes of a perfect square, that I can use to draw on.

While there are many methods deep inside my project, there is one method that no Java code can live without. It is the only method used in the main(): *launch*(args);.

```
public class Simon_4_MethodMadness extends Application {

    public static void main(String[] args) {
        launch(args);
    }
```

This is because methods are drawn on a canvas and not printed as words or numbers. In the JavaFX program, I used modifiers, like private and public. The private method *private void* drawBody(GraphicsContext gc) { is called in the public method *public void* start(Stage primaryStage) {. I also use class constructors, which in other words, is like the name of the modifier. For example, I used *drawFeet* and *drawBody.* I named the by what the class will do. These methods created this image.



Hunter Simon
Period 4

I believe my penguin is magnificent. I think I did the best I could have done.