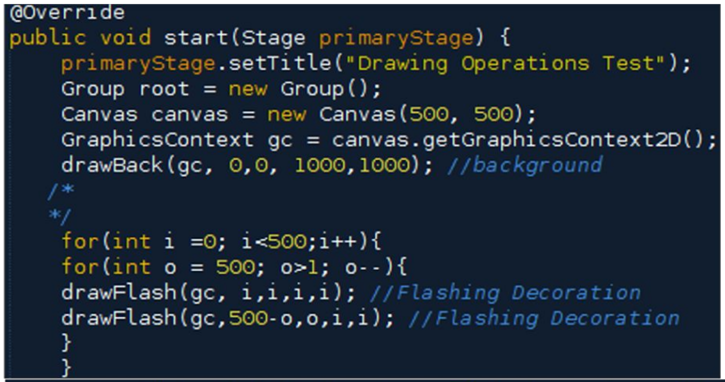
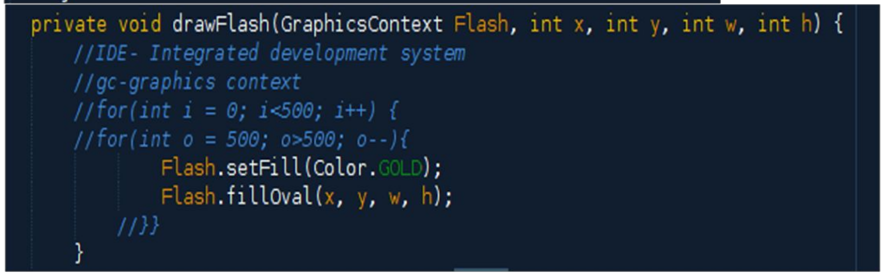
Challenge-1 Essay

My project, “Renard_7_D221”, creates a picture using methods. My code draws a sword that is implanted into the ground, similar to the excalibur, and draws a star-like figure in the background. In my opinion my project is more flat that most others, because of the lack of detail on certain objects. That being said, adding more detail past what i had already done would not have had much to do with coding, but more with copying and pasting things that i had already done. My program evidences principles of encapsulation because *private void drawName(gc);* public method is calling a private method *private void drawName(GraphicsContext gc) {.*

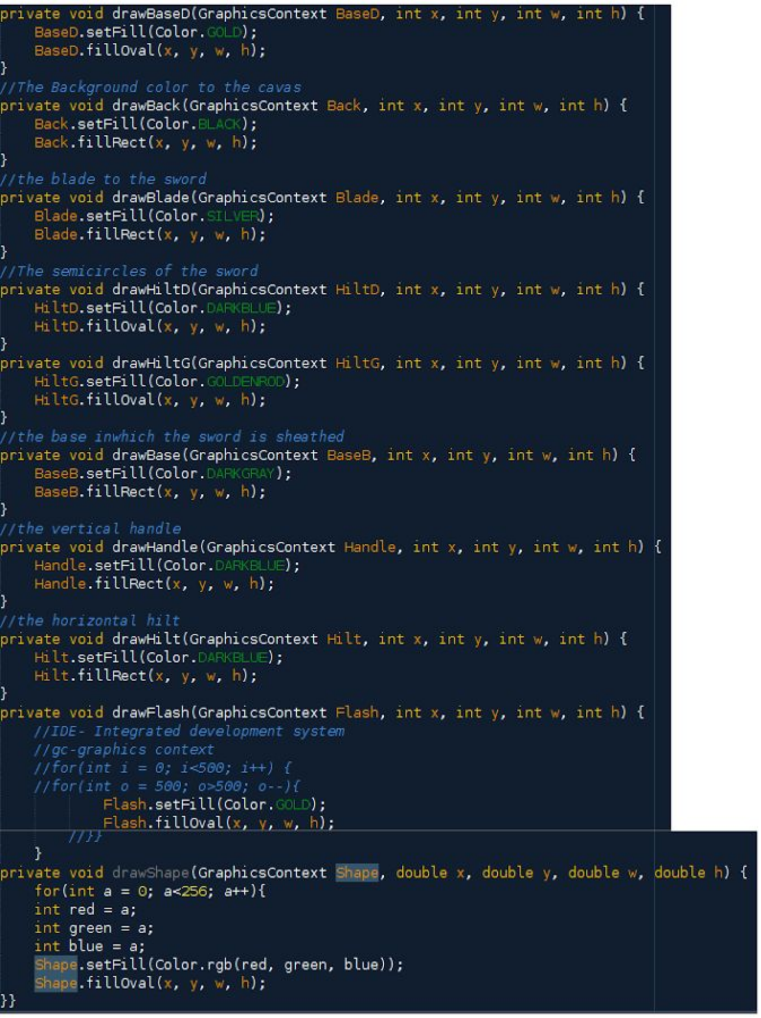


```
@Override
public void start(Stage primaryStage) {
    primaryStage.setTitle("Drawing Operations Test");
    Group root = new Group();
    Canvas canvas = new Canvas(500, 500);
    GraphicsContext gc = canvas.getGraphicsContext2D();
    drawBack(gc, 0,0, 1000,1000); //background
   /*
   */
    for(int i =0; i<500;i++){
    for(int o = 500; o>1; o--){
    drawFlash(gc, i,i,i,i); //Flashing Decoration
    drawFlash(gc,500-o,o,i,i); //Flashing Decoration
    }
    }
```



```
private void drawFlash(GraphicsContext Flash, int x, int y, int w, int h) {
    //IDE- Integrated development system
    //gc-graphics context
    //for(int i = 0; i<500; i++) {
    //for(int o = 500; o>500; o--){
            Flash.setFill(Color.GOLD);
            Flash.fillOval(x, y, w, h);
        //}}
    }
```

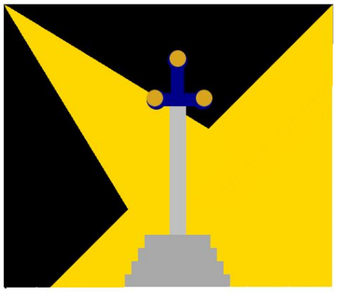
The methods I used are:



```
private void drawBaseD(GraphicsContext BaseD, int x, int y, int w, int h) {
    BaseD.setFill(Color.GOLD);
    BaseD.fillOval(x, y, w, h);
}
//The Background color to the cavas
private void drawBack(GraphicsContext Back, int x, int y, int w, int h) {
    Back.setFill(Color.BLACK);
    Back.fillRect(x, y, w, h);
}
//the blade to the sword
private void drawBlade(GraphicsContext Blade, int x, int y, int w, int h) {
    Blade.setFill(Color.SILVER);
    Blade.fillRect(x, y, w, h);
}
//The semicircles of the sword
private void drawHiltD(GraphicsContext HiltD, int x, int y, int w, int h) {
    HiltD.setFill(Color.DARKBLUE);
    HiltD.fillOval(x, y, w, h);
}
private void drawHiltG(GraphicsContext HiltG, int x, int y, int w, int h) {
    HiltG.setFill(Color.GOLDENROD);
    HiltG.fillOval(x, y, w, h);
}
//the base inwhich the sword is sheathed
private void drawBase(GraphicsContext BaseB, int x, int y, int w, int h) {
    BaseB.setFill(Color.DARKGRAY);
    BaseB.fillRect(x, y, w, h);
}
//the vertical handle
private void drawHandle(GraphicsContext Handle, int x, int y, int w, int h) {
    Handle.setFill(Color.DARKBLUE);
    Handle.fillRect(x, y, w, h);
}
//the horizontal hilt
private void drawHilt(GraphicsContext Hilt, int x, int y, int w, int h) {
    Hilt.setFill(Color.DARKBLUE);
    Hilt.fillRect(x, y, w, h);
}
private void drawFlash(GraphicsContext Flash, int x, int y, int w, int h) {
    //IDE- Integrated development system
    //gc-graphics context
    //for(int i = 0; i<500; i++) {
    //for(int o = 500; o>500; o--){
            Flash.setFill(Color.GOLD);
            Flash.fillOval(x, y, w, h);
        ///}}
    }
private void drawShape(GraphicsContext Shape, double x, double y, double w, double h) {
    for(int a = 0; a<256; a++){
    int red = a;
    int green = a;
    int blue = a;
    Shape.setFill(Color.rgb(red, green, blue));
    Shape.fillOval(x, y, w, h);
}}
```

The values passed through each method like *Hilt.fillRect(x,y,w,h);* are variables that will become coordinates/sizes for the shape that method creates. The specifications of the method are set whenever i use the method in the main class, this is so that i won't have to have the same method twice, but with four different numbers. Methods such as *Hilt.setFill(Color.DARKBLUE);*
pass a color instead of an integer. This method specifies the color that the 'Hilt' of the sword will be. The colors that I used are all imported using the command *import javafx.scene.paint.Color;.* The pictures are drawn onto a black 500x500 canvas so that the golds and blues pop more.

In this program, i used access modifiers such as private and public. Alongside these, i use class constructors, such as *drawHilt* and *drawBlade* which are used within their respective methods to define what the following commands will use. I named them like these so that i could easily come back to this project down the road long after i have forgotten what they do and be able to remember what they do. All these methods put together create this picture:



Although my image was not a complex or detailed at those made by my peers, the creation of it has allowed me to become much more proficient in putting together methods to achieve a goal. At the start of the project I was barely dragging along, slowly tinkering my way through my problems. By the end though, I was able to think up a shape that i wanted to make and implement it in mere minutes. Whenever I came across a problem I couldn’t work my way through, I would either ask my peers, or try my hand at stack overflow to see if the solution already exists.