Armand Sandiu
Eric Navarro

Report Lab4

The aim of this lab is the creation of some new classes while we recycle some of the previous labs. For example, we have to create the new abstract class entity, where the class Region inherits from there. But first, we have to change the class Point and create a class Vector because some classes need vector instead of points. For this, given two points, with a simple function we a vector.

```
public Vector difference( Point P2){
    double Xdifference= this.X -P2.getX();
    double Ydifference= this.Y- P2.getY();

    Vector v = new Vector(Xdifference,Ydifference) ;
    return v;
}
```

Then in class point the only thing we have to implement apart from a different method is the method move, that practically all our polygons will use to move.

```
public void move( double disX, double disY ){
    this.X += disX;
    this.Y += disY;
}
```

.
For entities and regions, we have some abstract methods that all types of polygons will share. As in the previous lab, the classes ellipsoidal and poligonal do the areas differently. Also the function translate is different because the ellipsoidals just move the center point but the polygonal must move all the points. It happens the same with the function draw because thanks to graphics class, we have different functions such as draw oval and draw polygon. For the function is point inside we compute with no trouble the formulas that are given to us.
e

```
g.setColor(fillcolor);
g.fillOval( (int) this.c.getX(), (int) this.c.getY(), (int) r1, (int) r2 );
g.setColor(lineColor );
g.drawOval((int) this.c.getX(), (int) this.c.getY(), (int) this.r1, (int) this.r2);
```

vs:

```
G.setColor(fillcolor);
G.fillPolygon( x, y,index );
G.setColor(lineColor);
G.drawPolygon( x, y, index );
```

Armand Sandiu
Eric Navarro

For the last subclasses that represent especifics polygons, we just have to call the supers of parents classes and pass the points that are needed for each figure. The example of triangular class:

```
public TriangularRegion(Color InitilineColor, Color Initfillcolor,Point p1, Point p2, Point p3) {
    super(InitilineColor,Initfillcolor ,new LinkedList<Point>(Arrays.asList(p1,p2,p3)));

}
```

The implementation of text and line was easy in sense that for the draw we just need the appropriate functions.

Last part of the lab was the hardest one because we had to implement our color class. For this we create two classes more that are HSV and RGB to pass to one model to the other. RGB receives the points we introduce on color class and translate them to pass it to HSV. For this we casted the color points into int(were in double type).

Then, we create a test class for Entity_drawers, and as is said on the statement, we create a entities drawers instance and we pass later the polygons we want with each respective points.