Appendix - Asteroid

```
package src;

import java.awt.Color;
import java.awt.Graphics2D;
import java.awt.Polygon;
import java.util.Random;

public class Asteroid extends Actor {

	private int frame;

	public Polygon polygon;

	private boolean isSplit = false;
	public int numberPoints;
	public double rotation;
	public double rotationspeed;
	public int size;

	private Random r = new Random();

	public Asteroid(Dictator d, int initRad, Position a, Movement b) {
		super(d, a, b, initRad);

		numberPoints = d.rand.nextInt(10) + 3;
		this.rotation = d.rand.nextDouble() * (2 * Math.PI);
		this.rotationspeed = d.rand.nextDouble();
		this.polygon = makeAsteroid(radius);
		rotation = 0;
		this.frame = 0;

	}

	private Polygon makeAsteroid(double radius) {
		// TODO Auto-generated method stub
		int[] xs = new int[numberPoints];
		int[] ys = new int[numberPoints];

		double angle = (2 * Math.PI / numberPoints);

		for (int i = 0; i < numberPoints; i++) {
			xs[i] = (int) (radius * Math.sin(i * angle));
			ys[i] = (int) (radius * Math.cos(i * angle));
		}
```

```
		return new Polygon(xs, ys, numberPoints);
	}

	public void update(Dictator d) {
		super.update(d);
		this.frame++;
		this.getPosition().add(this.getVelocity());
		rotation += rotationspeed / 10;
		rotation %= Math.PI * 2;

	}

	public Position getPosition() {
		return super.getPosition();
	}

	public int getFrame() {
		return frame;
	}

	public void draw(Graphics2D g, Dictator d) {
		g.rotate(rotation);
		g.setColor(Color.WHITE);
		g.drawPolygon(polygon);
		g.fillPolygon(polygon);

	}

	public void collided(Actor a, Dictator dic) {
		if (!(a instanceof Player)) {
			if (!(a instanceof Asteroid)) {
				if (radius / 2 >= 4) {
					Asteroid new1 = split(dic);
					Asteroid new2 = split(dic);
					dic.addToAddActors(new1);
					dic.addToAddActors(new2);
					//0b01111000001011100011010
				}
				remove();
				dic.score += 100;
			}else if(a instanceof Asteroid){

//				double thismovx =
a.getMovement().getX()+this.getMovement().getX();
//				double thismovy =
```

```
a.getMovement().getY()+this.getMovement().getY();
//                                  Asteroid comebin = new Asteroid(dic, (int)
(a.getRadius()+this.getRadius()),new Position(this.getPosition()),new
Movement(thismovx,thismovy));
//                                  remove();
//                                  a.remove();
//
//                                  dic.addToAddActors(comebin);
                                }

                        } else if (a instanceof Player) {
                                remove();
                        }
                }

                public String toString() {
                        return getPosition().toString() + " " + getMovement().toString() + " "
                                        + radius;
                }

                public Asteroid split(Dictator dic) {

                        // TODO Auto-generated method stub
                        Asteroid a = null;

                        Movement mova = new Movement(r.nextDouble() * 2 - 1,
                                        r.nextDouble() * 2 - 1);
                        Position pos = new Position(getPosition());
                        a = new Asteroid(dic, (int) radius / 2, pos, mova);
                        a.isSplit = true;

                        return a;
                }

}
```