/\*

\* 中子模型

\*

\*/

package application.Model;

import javafx.scene.paint.Color;

import javafx.scene.paint.CycleMethod;

import javafx.scene.paint.RadialGradient;

import javafx.scene.paint.Stop;

import javafx.scene.shape.Circle;

public class NeutronModel {

private int no;

private int Xvelocity; //横坐标速度

private int Yvelocity; //纵坐标速度

private Circle circle; //用圆来描绘中子

private RadialGradient gradient;//填色

private int XCoordinates; //横坐标

private int YCoordinates; //纵坐标

public NeutronModel() {//初始化函数

//设置横纵坐标速度

Xvelocity=(int) (Math.random()\*10)+1;

Yvelocity=(int) (Math.random()\*10)+1;

while(Yvelocity==0||Xvelocity==0) {

Xvelocity=(int) (Math.random()\*10);

Yvelocity=(int) (Math.random()\*10);

}

//圆的大小

circle = new Circle(40,40,10);

//设置渐变颜色

gradient = new RadialGradient( 0, .1, 40, 40, 10, false,

CycleMethod.NO\_CYCLE, new Stop(0, Color.WHITE), new Stop(1, Color.RED));

circle.setFill(gradient);

}

public void setno(int no) {

this.no=no;

}

public int getno() {

return no;

}

public int getXVelocity() {

return Xvelocity;

}

public int getYVelocity() {

return Yvelocity;

}

public Circle getCircle() {

return circle;

}

public void setCircle(Circle circle) {

this.circle=circle;

}

public void setXCoordinates(int Coordinates) {

this.XCoordinates=Coordinates;

circle.setCenterX(Coordinates);

}

public void setYCoordinates(int Coordinates) {

this.YCoordinates=Coordinates;

circle.setCenterY(Coordinates);

}

public void setXvelocity(int velocity) {

this.Xvelocity=velocity;

}

public void setYvelocity(int velocity) {

this.Yvelocity=velocity;

}

public int getX() {

int X;

double xMin = circle.getBoundsInParent().getMinX();

double xMax = circle.getBoundsInParent().getMaxX();

X=(int)((xMax+xMin)/2);

// TODO Auto-generated method stub

return X;

}

public int getY() {

int Y;

double yMin = circle.getBoundsInParent().getMinY();

double yMax = circle.getBoundsInParent().getMaxY();

Y=(int)((yMax+yMin)/2);

return Y;

}

public int getSize() {

// TODO Auto-generated method stub

return 10;

}

}