/\*

\* 在没有原子核的情况下，这些中子在封闭空间中自由运行（不用互相撞击）

\*/

package application.File;

import application.Model.\*;

import javafx.animation.Animation;

import javafx.animation.KeyFrame;

import javafx.animation.Timeline;

import javafx.application.Application;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

import javafx.scene.Group;

import javafx.scene.Scene;

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;

import javafx.stage.Stage;

import javafx.util.Duration;

public class NeutronShow extends Application {

Stage stage=new Stage();

private int NeutronCount=5; //设置有五个中子

private NeutronModel []Neutron=new NeutronModel[NeutronCount];//中子模型

@Override

public void start(Stage primaryStage){

int i;

for(i=0;i<NeutronCount;i++)

Neutron[i] =new NeutronModel();

primaryStage.setTitle("中子自由运行动画显示");

Scene scene = new Scene(new Group(), 450, 350);

Group root = (Group) scene.getRoot();

try {

//设置时间轴

Timeline []time=new Timeline[NeutronCount];

//设置关键帧

KeyFrame []moveNeurtion=new KeyFrame[NeutronCount];

for(i=0;i<NeutronCount;i++) {

//初始化关键帧

time[i]=new Timeline();

time[i].setCycleCount(Animation.INDEFINITE);//indefinite:不确定的。

final int j=i;

moveNeurtion[i] = new KeyFrame(Duration.seconds(.01500),

new EventHandler<ActionEvent>(){

@Override

public void handle(ActionEvent event) {

//碰到边界改变方向

changevelocity(Neutron[j],j);

}

});

time[i].getKeyFrames().add(moveNeurtion[i]);

time[i].play();

root.getChildren().add(Neutron[i].getCircle());

}

primaryStage.setScene(scene);

primaryStage.show();

}catch (Exception e) {

e.printStackTrace();

}

}

//通过函数获得控件的边界位置，以此判断控件的坐标，是否需要换速度方向

public void changevelocity(NeutronModel Neutron,int i) {

Circle circle=Neutron.getCircle();

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

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

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

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

if(xMin<0 || xMax>450){

Neutron.setXvelocity(Neutron.getXVelocity()\*-1);

}

if (yMin < 0 || yMax > 350) {

Neutron.setYvelocity(Neutron.getYVelocity()\*-1);

}

//改变了中子的坐标后将新的中子返回给模型

circle.setTranslateX(circle.getTranslateX() + Neutron.getXVelocity());

circle.setTranslateY(circle.getTranslateY() + Neutron.getYVelocity());

Neutron.setCircle(circle);

}

public void showWindow() throws Exception {

start(stage);

}

}