# **操作系统课程**

# **设计报告**

**设计类型：多线程的实验和运用**

**设计题目： 小球大乱碰**

**系别： 计算机科学与技术**

**年级专业： 计算机**

**学号： 201520721051**

**姓名： 向波**

**指导老师： 张一**

[操作系统课程 1](#_Toc4494)

[设计报告 1](#_Toc20971)

[1、设计目的 3](#_Toc3834)

[1.1设计思路 3](#_Toc8651)

[1.2设计原理 3](#_Toc32273)

[1.3设计流程图 3](#_Toc4867)

[4](#_Toc24336)

[2、 程序使用说明 4](#_Toc430)

[2.1程序演示 4](#_Toc20274)

[2.2代码说明 6](#_Toc10626)

[3、 课程总结 7](#_Toc26926)

[4参考文献 7](#_Toc14534)

## 1、设计目的

通过设计一个图形界面，使多个小球在界面中坐规则运动，产生碰撞，每个小球分配一个线程，达到控制多线程的目的。

### 1.1设计思路

首先生成一个图形界面，大小为800\*600，使多个小球在其中不会感觉拥挤，然后把这个界面继承在JPanel面板上，提高画面流畅度，解决JFrame的闪屏问题，然后将小球建立为数组保存，为每一个小球分配一个大小为Ball函数的空间，小球的颜色与位置随机生成，为每一个小球分配一个线程，使他们能分别运动，然后写一个碰撞函数，使小球发生碰撞时能够按照物理定律进行反弹。

### 1.2设计原理

要使多个小球能够在界面中运动，首先得编码一个线程的类，然后多个小球以此类开辟线程、保存数据（半径、速度、x坐标、y坐标、颜色、在哪个画板上）

要实现小球的碰撞，要分别实现小球的边界碰撞和相互碰撞。

首先边界碰撞在Ball函数中实现，检测小球在下次运动时是否会达到边界，如果达到左右边界，则将小球的x轴上的速度取反，如果达到上下边界，则将小球的y轴上的速度取反。小球就能在边界进行碰撞一样的运动了。

然后相互碰撞是利用数学公式计算出小球之间的相互距离，用一个double类型的二维数组保存，然后在小球每次运动后检测是否发生相互碰撞，当时，说明小球应该发生碰撞，然后将两个小球的速度交换。

### 1.3设计流程图

本设计流程图展示了小球的某一次碰撞：

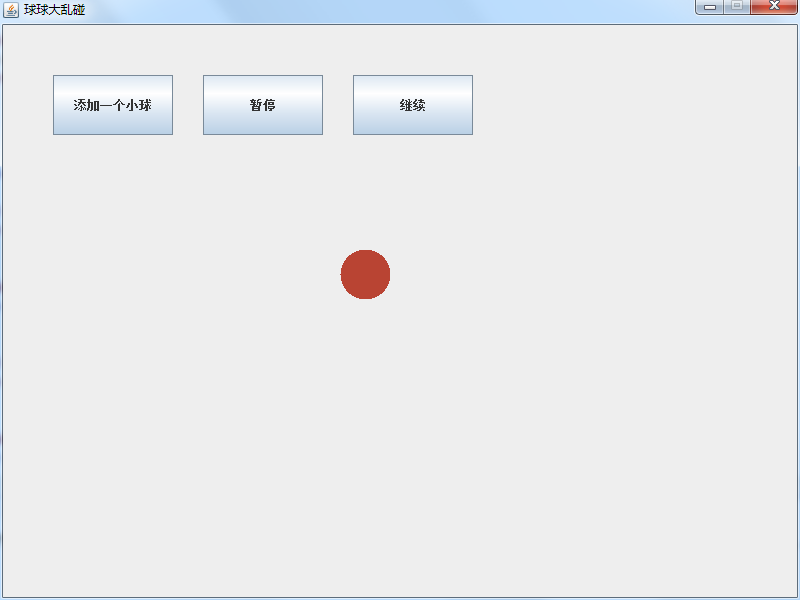
## 未命名文件

## 程序使用说明

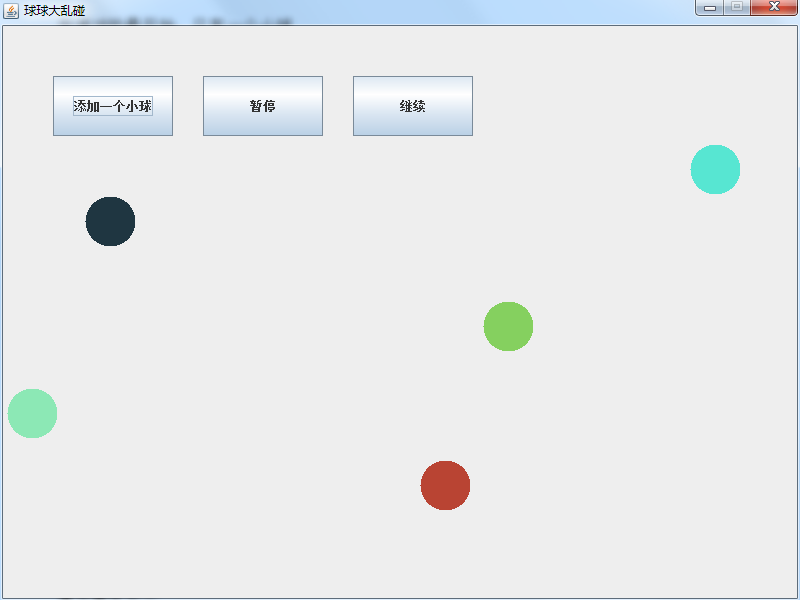
点击 求求大乱碰.jar 应用打开程序进行运行

### 2.1程序演示

在游戏的最开始，只有一个小球



可以点击 添加一个小球 按钮 增加一个小球，每一个小球代表了一个线程，小球的数量设置为最多五个



点击界面中的暂停按钮，可以使所有的小球停止，点击继续按钮，可以使小球重新运动

### 2.2代码说明

1. 线程的暂停：

**synchronized** (**this**) {

**if** (suspended) {

**try** {

**this**.wait();

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

当我的暂停按钮按下时，suspended会变为true，然后运行以上程序，使每个小球的线程暂停

**public** **void** setSuspend(**boolean** suspend) {

**if** (!suspend) {

**synchronized** (**this**) {

**this**.notifyAll();

}

}

**this**.suspended = suspend;

}

当按下继续时，唤醒所有线程

**public** **void** addball(**int** ballnum){

ball[ballnum]=**new** Ball(**new** Color(r.nextInt(255),r.nextInt(255),r.nextInt(255)),5,5,r.nextInt(750),r.nextInt(550),25,**this**,**false**);

ball[ballnum].start();

}

按下添加小球按钮时，新建一个Ball类，将小球的颜色、大小，速度，都保存下来，并且启动这个小球的线程

**public** **void** impact(){

**int** t;

**double** dis[][]=**new** **double**[6][6];

**double** X1,X2,Y1,Y2;

**for**(**int** i=1;i<=ballnum;i++){

**for**(**int** j=1;j<=ballnum;j++){

X1=ball[i].getW()+ball[i].getR();

X2=ball[j].getW()+ball[j].getR();

Y1=ball[i].getH()+ball[i].getR();

Y2=ball[j].getH()+ball[j].getR();

dis[i][j]=Math.*sqrt*((X2-X1)\*(X2-X1)+(Y2-Y1)\*(Y2-Y1));

}

}

**for**(**int** i=1;i<=ballnum;i++){

**for**(**int** j=i+1;j<=ballnum;j++){

**if**(dis[i][j]<=ball[i].getR()+ball[j].getR()){

t=ball[i].getX();

ball[i].setX(ball[j].getX());

ball[j].setX(t);

t=ball[i].getY();

ball[i].setY(ball[j].getY());

ball[j].setY(t);

}

}

}

}

这是小球的碰撞函数，用一个dis[][]的浮点型二维数组保存小球之间的距离，然后判断每个小球是否发生碰撞，如果发生碰撞，交换速度和速度方向。

## 课程总结

此次课程让我加深了对于多线程的控制，深入地了解了多线程在java以及c++中的一些运用，而且我学习到了一些关于线程的操作，比如可以在线程的类中加入一个函数类型的变量，这样就可以在线程的类中进行repaint，省去了很大的功夫，而且我深入掌握到了一些碰撞在编程语言中运用的方式，比如当两个小球发生碰撞，先用一个int类型的变量接收小球a的速度，然后将小球a的速度赋值为小球b的速度，再将小球b的速度赋值为记录的变量，这与冒泡算法在交换时的运用是一样的。我认为这节课程总体难度比较低，但我们可以将自己的程序不断地开发，加深理解，最终以多线程为基础，开发出一个不错的应用。

## 4参考文献

<http://www.jb51.net/article/49494.htm> 会反弹的小球示例（一个小球的反弹运动）