小球碰撞模型的物理分析、算法设计和基于OpenGL的可视化实现

小组成员：朱宝林、尤比佳、林佳诚、张宏彬

指导老师：黄罗华

摘要：

关键词：

# 绪论

## 研究背景

经过几个世纪的发展，数学、物理等基础学科理论已日益完善。随着研究内容越来越深入，其高度抽象的公式无法直观理解，复杂的计算难以靠人力来完成。现代科学研究常以数学、物理等基础学科作为理论基础，而将计算机作为计算和可视化工具。此次研究性学习，我们选取物理学的经典模型——小球碰撞为研究对象，进行物理理论分析，并尝试在计算机上模拟实现以此模型为基础的一些应用场景。

小球碰撞模型在物理研究中具有重要意义。物理学的一个重要方法是理想化模型。当物体自身形状可以忽略时，常常将其简化为小球或质点进行处理。借助小球碰撞模型，物理学家对理想气体的微观运动进行了深入探索。小球碰撞模型还在化学反应动态过程、元素相变、最密堆积问题、天体物理等领域具有广泛应用。

在现代科学研究中，计算机已经成为不可或缺的一环。计算机凭借其强大的性能和高效的算法，能够快速处理海量数据并得到准确的结果。利用可视化技术，研究人员得以实时观察数据情况，获得直观感受

## 研究目的与意义

通过此次研究，我们希望深入探索小球碰撞这一经典模型，学习物理研究的基本方法；探索在计算机上模拟物理过程，学习使用算法处理大量数据；学习使用图形库将研究结果可视化；学习运用科学研究方法探索、分析并解决问题，实际体验科学研究过程。这将极大地培养我们的和科学研究能力，为我们将来从事科学研究工作打下坚实基础。

## 研究内容

物理：刚性小球碰撞模型：物体模型、状态模型、过程模型

算法：碰撞检测、事件处理（排序）、驱动模拟

可视化：OpenGL库的使用

## 研究方法

实验法、文献研究法、模型方法、信息研究方法

# 物理分析

## 物理建模

为了研究小球碰撞模型，我们需要设计一个的物理系统作为基础。考虑到需要研究的问题，系统中有应当小球和一些固定的障碍物。我们需要建立实体物理模型、状态物理模型和过程物理模型这三类模型。

实体物理模型

该理想化系统中，所有物体均为刚体，不考虑摩擦。

小球：质量m，半径r，恢复系数e。

平面：恢复系数e，位置r，法向量n 。

面片：恢复系数e。通过记录面片的各个顶点来描述该面片。

长方体：恢复系数e。通过记录它的各个顶点来描述该长方体。

状态物理模型

在系统中只有小球是运动的，其它物体均固定。

小球具有的状态量有位置r，速度v，加速度a。它们都是关于时间t的函数。

## 过程分析

### 碰撞判断和时间计算

#### 球与球之间的碰撞

球体占有到球心一定距离内的空间。因此，可以利用空间中两点距离公式确定两球是否发生相撞。

设空间中两球位置分别为 和 ，两球速度为 和 。若两球在t时刻相撞，则有：

|  |  |
| --- | --- |
|  | (2.1) |

代入坐标展开，用每对量的差表示可得一元二次方程：

|  |  |
| --- | --- |
|  | (2.2) |

容易计算方程判别式，若 得到方程两根：

|  |  |  |
| --- | --- | --- |
|  |  | (2.3) |

对方程根的情况分类讨论，可知：

1. 若方程两根均小于0，则两球不会发生碰撞。（负根说明：若时间倒退，两小球可以发生碰撞）
2. 若方程两根不都小于0，则两球发生碰撞的时间是最小的非负实根

### 碰撞处理

任何维度的小球碰撞均可转化为沿球心连线方向的一维碰撞，在其他方向上运动状态不变。这是因为小球碰撞时，接触力的方向垂直于小球接触点的切面，也就是从接触点指向球心。

假设t时刻两小球发生碰撞，取该时刻小球位置矢量相减并归一化得到球心连线方向矢量

|  |  |  |
| --- | --- | --- |
|  |  | (2.4) |

将两小球的速度与方向矢量点乘得到速度在球心连线方向的分量

|  |  |  |
| --- | --- | --- |
|  |  | (2.5) |

在球心连线方向上（一维空间）处理小球碰撞，应用《力学》中联立动量、动能守恒方程得到的一维碰撞解

|  |  |  |
| --- | --- | --- |
|  |  | (2.6) |

计算该方向上速度变化量，将变化量乘以方向向量回到三维空间中，再叠加回原速度

|  |  |  |
| --- | --- | --- |
|  |  | (2.7) |

从而完成了一次小球碰撞的计算

### 引入恢复系数

## 应用

台球

布朗运动

# 面向对象程序设计

在这一章，我们主要使用UML (Unified Modeling Language, 统一建模语言)来描述我们的程序设计过程。

## 面向对象分析

为了使用计算机实现对上述物理过程的运算，并将其运用到更多模型中，我们将其抽象为程序中的对象。一个简单的碰撞系统应当可以输入数据，根据所需的时间更新系统状态，显示系统当前的状态。图3-1给出了碰撞系统的用户视图。

按第二章物理分析的思路，我们为碰撞系统创建四个实体类和一个事件类。实体类应当保存实体的状态，并提供对实体的一些操作；事件类保存事件的信息，并提供对事件的处理（在实现中，处理会转交给具体的实体）。图3-2给出了该碰撞系统的类图。

3‑1碰撞系统的用户视图

一次碰撞事件中必定有一个小球对象参与。因此，碰撞处理的操作既可以全部交给在小球实体类（即实现小球与各个实体的碰撞处理），也可以交给每个类处理（即实现各个实体与小球的碰撞处理）。我们选择后者，因为系统中必定存在小球，而不一定存在其他实体。这样做可以提高实现效率，也进一步加强了类之间的封装性。

## 面向对象设计

分析完成后，我们需要详细描述类的细节，列出它们的属性和方法的细节。图3-3给出了碰撞系统设计中使用的3个类的细节。

## API

依照上一节，我们定义了以下抽象数据类型及其API。

小球和墙类：储存数据，并具体处理相关操作。为了简化运算，我们在Ball类中使用了GLM数学库中的向量vec3类表示小球在世界空间中的位置和速度。

事件类：负责储存事件信息，并负责处理事件、更改小球和墙的数据

# 算法分析和实现

## 实现

### 事件

一次碰撞事件至少应当包含：发生碰撞的两个物体的信息，发生碰撞的时刻

事件类型：在我们构建的物理系统中，碰撞有小球之间的碰撞和小球和墙之间的碰撞，我们分别使类内bool类型成员type为0或1来表示。

事件有效：我们提出两种方法：①处理事件时判断有效性：在类内创建小球碰撞次数的记录。显然，如果在处理该事件时小球的累积碰撞次数发生了变化，即与事件中的记录不符，则在此事件发生前小球必定已经发生了碰撞，因此可以判断事件是否有效。②处理碰撞时删去无效事件：碰撞后，小球的运动轨迹发生改变，与该小球有关的事件均失效，因此可以在每次碰撞后标记或删去无效事件。我们预测：方法①具有更高的效率，方法②可以在一定程度上减少内存使用冗余。

事件时间：在优先队列中，事件按绝对时间进行排序，即从t=0的初状态开始的时间。timeToCollision方法仅能得到事件发生距离当前时刻的时间，在事件中记录时间时应当加上当前时刻的时间。

### 时间

## 算法探究1：时间驱动和事件驱动

## 算法探究2：包围盒检测

## 算法探究3：优先队列的长度

若系统较为密集，则短时间内可能发生较多碰撞事件。为了防止内存溢出，又要保证效率最大，应当合理设计优先队列的长度

## 实验1：回退

# OpenGL可视化

## 3D模型构造

### 平面绘制

在OpenGL中，一个平面应当具备：顶点坐标、法向量、纹理坐标

绘制平面的步骤：

## 动画：相对时间和绝对时间

由于使用了OpenGL动画，动画的绘制方式是双缓冲，逐帧绘制。为了使模拟尽可能贴近实际，考虑绘制和事件处理时间的关系就尤为重要。

我们以初始化（第一次绘制）作为时间起始点，绘制一条时间轴。每次处理的流程如图。相应时间在时间轴上标出：

# 模拟实验

## 台球

## 拥挤的人群

## 布朗运动

## 加入参数：温度

# 参考文献

参考文献1

参考文献2

# 样式参考集

应用标题：首先应用样式，然后点击列表配置

一个标准表格

元素中间对齐，使用独立样式，释放行距限制，右侧右对齐

|  |  |  |
| --- | --- | --- |
|  |  | （1.7） |