Java大作业

----桌上球类游戏设计

课程：Java技术与应用

时间：2016秋季

学生：何宜晖 计算机46 2140504137

谢茹吉 自动化46 2140504148

指导老师：卫颜俊老师

Contents

[1 开发背景与技术 2](#_Toc468508904)

[2 系统目标 3](#_Toc468508905)

[3 系统分析与设计 3](#_Toc468508906)

[3.1 人机界面设计 3](#_Toc468508907)

[3.2 类的设计 4](#_Toc468508908)

[3.3 算法的设计 6](#_Toc468508909)

[3.4 实验数据与测试用例准备 7](#_Toc468508910)

[4 使用说明书 7](#_Toc468508911)

[5 进度与人员安排 11](#_Toc468508912)

[6 开发总结 11](#_Toc468508913)

[7 附件程序 11](#_Toc468508914)

[8 附件数据库 11](#_Toc468508915)

[9 附件文档 11](#_Toc468508916)

# 开发背景与技术

面向对象程序设计作为一门软件设计的课程，具有极强的实践性，要求学生具备灵活应用理论知识的能力及面向对象程序设计技能的基础。通过游戏开发，学生能了解java面向对象的设计方法与技巧，有效地、深刻地理解课程内容，体会理论、方法和设计原则，培养分析实际问题和解决问题的能力，具备使用面向对象程序设计开发工具设计实际系统的能力。还能够了解并通过使用eclipse，掌握一种可视化编程的方法，并通过游戏的开发加深对可视化编程的理解。同时，可以提高运用java编程语言解决实际问题的能力。 桌球游戏属于休闲类游戏，具有上手快、游戏时间短的特点，更利于用户进行放松休闲，为人们所喜爱。方便、快捷、操作简单，在休闲娱乐中占重要位置。开发出了计算机桌球游戏，以后不仅仅可以进行休闲，还能锻炼自己的智力和桌球技术，更加方便了人们的日常生活。

# 系统目标

1. 界面设计，程序界面具有基本菜单，正确的球台绘制。摆放球正确，开球前目标球排列为三角形,共5排,每排球数分别为一至5颗。第一排的1颗球置于‘置球点’,8号球位于第三排的中间位置,其它目标球全色和花色间隔开随意摆放,但必须彼此紧贴。白球位置可由玩家开局时自动摆放。
2. 开球实现球的运动符合物理规律且桌球滚动动画连贯，台球路线由两方面决定：（合理击球情况下）母球由球杆击打得出运动方向、白球由碰撞决定运动方向，子球由碰撞决定运动方向，球的x（横坐标），y（纵坐标）可以定位其当前位置，dx（横坐标变化量），dy（纵坐标变化量）可以确定其速度大小及方向，ddx，ddy可以确定球的加速度大小及方向
3. 实现合理的桌球规则，如主球入袋需重新摆放等。
4. 判断游戏是否结束。

# 系统分析与设计

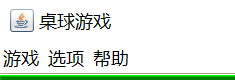
这次大作业，我们使用了在课上学习的几种技术，包括，Java线程，Swing，AWT，多媒体等等。以及利用了物理公式建立了简单的牛顿定律模型。

## 人机界面设计

人机界面主要包含两个方面，菜单，以及台球桌面。

**菜单**，在Menu.java中实现，使用了JFrame，用ActionListener来监听菜单事件。当有按键按下后执行相应的功能。

* 我们用JCheckBoxMenuItem 来方便调节游戏设置
* JMenuItem 则是普通的菜单选项
* JMenu 包含三个主要功能，游戏，选项，帮助



**台球桌**， 所有的界面都在paint中，

* 背景颜色直接采用 setBackground(**new** Color(0, 127, 0)); 来填成绿色。
* 台球桌，画两个嵌套的方框就比较像台球桌了，g.drawRect(Yinqing.tbx - 30, Yinqing.tby - 30, sx + 60, sy + 60); g.drawRect(Yinqing.tbx, Yinqing.tby, sx, sy);
* 球洞则在相应位置画圆即可 g.fillOval(p.x, p.y, p.size, p.size);
* 球也是一样的画圆，不过不同的是需要在表面写上不同的球编号 g.drawChars(n, 0, 1, (int) (b.x + 12), (int)(b.y + 17)); 球内部的图案有时候需要用到画长方型 g.fillRect((int) b.x + 11, (int) b.y, 8, r2);

**交互**，在台球游戏中，主要包含了鼠标点击，鼠标拖拽，鼠标释放的操作。

* 鼠标点击mousePressed，鼠标按下时，如果当前有的球还没停下，则禁止用户操作。
* 鼠标拖拽mouseDragged，拖拽的时候，需要重新计算出虚拟的辅助瞄准线。如果没有的可以跳过。
* 鼠标释放mouseReleased，鼠标释放时，需要根据力量的大小，确定白球的速度。

**声音效果**，声音效果控制，在球碰撞，球进洞，杆子释放的时候触发。在Sound。Java中，我们定义了声音枚举类型，如果当前正在播放音乐clip.isRunning()，则会停止clip.stop(); 重新播放音效clip.setFramePosition(0);clip.start();

* 球碰撞，球相互撞击的声音Sound.HIT.sound();
* 球进洞，进洞响声 Sound.SINK.sound();
* 杆子释放，发出撞球声音Sound.CLASH.sound();

## 类的设计

**主类**，Main。Java

程序入口，用于生成界面。

**new** Menu();

**球袋类， qiudai。Java**

**简单定义了球带大小，以及坐标。**

**public** **int** size=48;

**public** **int** x;

**public** **int** y;

**球类 Ball。Java**

**定义了球的名字，颜色，坐标，速度，加速度，标号，大小等。**

**public** String mingzi = "";

**public** Color yanse;

**public** **int** daxiao,bianhao;

**public** Point zhongxin;

**public** **double** x,y, dx,dy,ddx,ddy;

**public** **boolean** yidong = **false**, mouseDrag = **false**;

**界面类 Menu。Java**

**建立菜单，监听菜单事件，以改变游戏进行状态。**

**public Menu() 建立了菜单**

**private** **void** addMenuItems(JMenu menu, String items) 根据给定字符串，简便添加不同属性的菜单按钮。

**public** **void** actionPerformed(ActionEvent e) 用于监听菜单事件。与主类Game类相关，可改变主类状态。

**else** **if** (e.getActionCommand().equals("新游戏")) {

game.newGame();

**else** **if** (e.getActionCommand().equals("考虑摩擦"))

game.yinqing.f = !game.yinqing.f;

**else** **if** (e.getActionCommand().equals("辅助瞄准"))

game.yinqing.miaozhun = !game.yinqing.miaozhun;

**声音效果类，Sound.java**

**定义了播放声音的枚举。**

***HIT***("sounds/clash.wav"), ***SINK***("sounds/fall.wav"), ***CLASH***("sounds/hit.wav"); 指定三种声音文件的路径。

Sound(String s) { **用于读取声音文件**

**public** **void** sound() 方便其他类发生，且中断之前播放的音乐。

**物理效果类 Yingqing.Java**

**物理引擎类，继承了线程，因为我们需要实时，更新每个球的状态，判断球有没有碰撞，有没有进洞，有没有停下。同时，这个类也起到了根据牛顿第二定律，速度交换原理，计算球状态的作用。**

**public** Yinqing( Game game ) 初始化

**public** **static** **boolean** cCol(Ball i, Ball j ) 计算两个球有没有碰撞。

**public** **void** cF( Ball b) 计算摩擦力，加速度越大，摩擦力越大

**private** **void** dF( Ball c ) 根据摩擦力，更新当前球的状态

**private** **boolean** ifQiudai( Ball b ) 判断球是否落入球袋

**private** **void** ifBounds( Ball b) 判断球是否到达边界，如果是，运用反射定律

**public** **void** cPos() 更新球状态

**private** **void** cRd() 检查球是否停止

**public** **void** run() 线程，不断通过cPos计算球的当前位置

**游戏类，Game.java**

**游戏类实现了MouseListener, MouseMotionListener（监听鼠标事件）, Runnable（线程，由于需要不断绘图）**

**public** Game(Menu mymenu) 新建

**public** **void** newGame() 新游戏，当menu中点击新游戏时，调用cBalls，重新摆放球。

**public** **void** rdBall() 找到白球

**public** **int** fbaiidx(Vector<Ball> balls) 返回白球位置

**public** **void** createQiudais() 按照预先指定的位置，创建球袋实例

**public** **void** cBalls() 按照三角形摆放好球

**private** **void** pball(Graphics g, Ball b) 根据给定的球实例，画出球

**public** **void** adFall(Ball c) 对于掉落的球，画在窗口下方

**public** **void** paint(Graphics g) override，绘制台球桌，球，球洞等。

**public** **void** mouseClicked(MouseEvent e) 空，必须实现

**public** **void** mousePressed(MouseEvent e) 鼠标按下的时候，

**public** **void** mouseDragged(MouseEvent e) 改变辅助瞄准

**public** **void** mouseReleased(MouseEvent e) 计算白球初速度，加速度

**public** **void** mouseEntered(MouseEvent e) 空，必须实现

**public** **void** mouseExited(MouseEvent e) 空，必须实现

**public** **void** mouseMoved(MouseEvent e) 不让球超出台球桌

**public** **void** run() 线程，不断重绘台球桌

## 算法的设计

**并行双线程**，

我们出于速度考虑，使用两个线程，由于画图线程略慢一些，所以需要和物理引擎分开。物理引擎可以实时计算球的轨迹。

**物理引擎**，

主要是将物理定律用Java实现。

例如碰壁时，速度反向，加速度取正向。

c.dy = -Math.abs( c.dy );

c.ddy = Math.abs( c.ddy );

例如两球碰撞时，速度置换：

kba = (dx \* a.dx + dy \* a.dy) / ds;

kaa = (dx \* a.dy - dy \* a.dx) / ds;

kab = (dx \* b.dx + dy \* b.dy) / ds;

kbb = (dx \* b.dy - dy \* b.dx) / ds;

a.dy = kab \* dy + kaa \* dx;

a.dx = kab \* dx - kaa \* dy;

b.dy = kba \* dy + kbb \* dx;

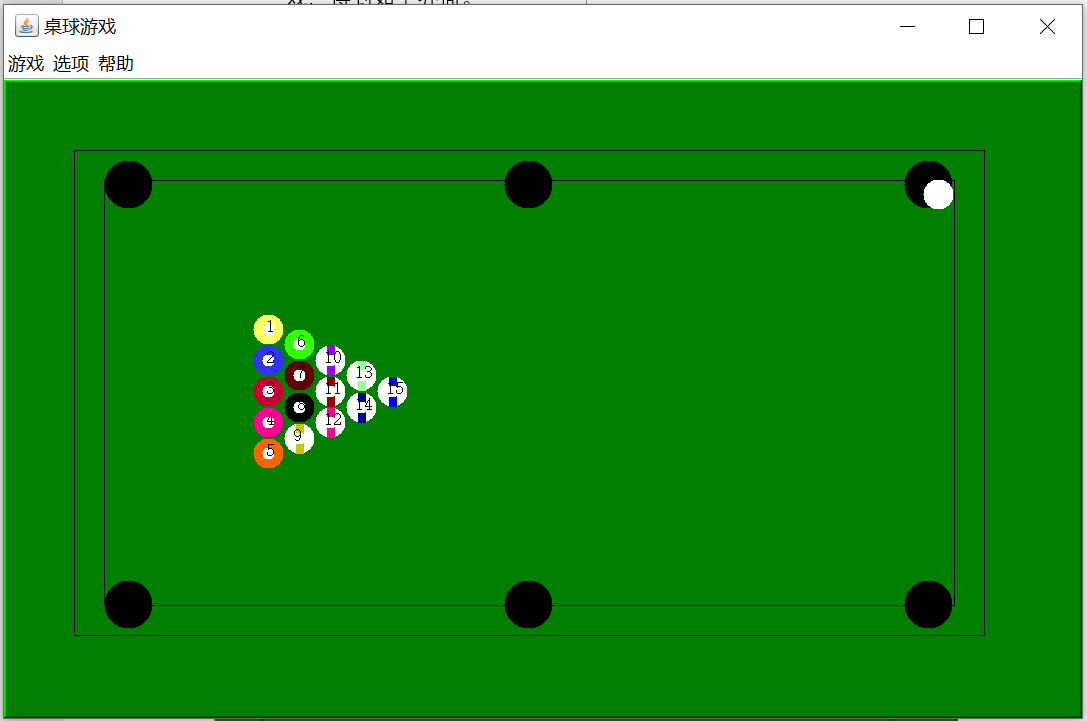
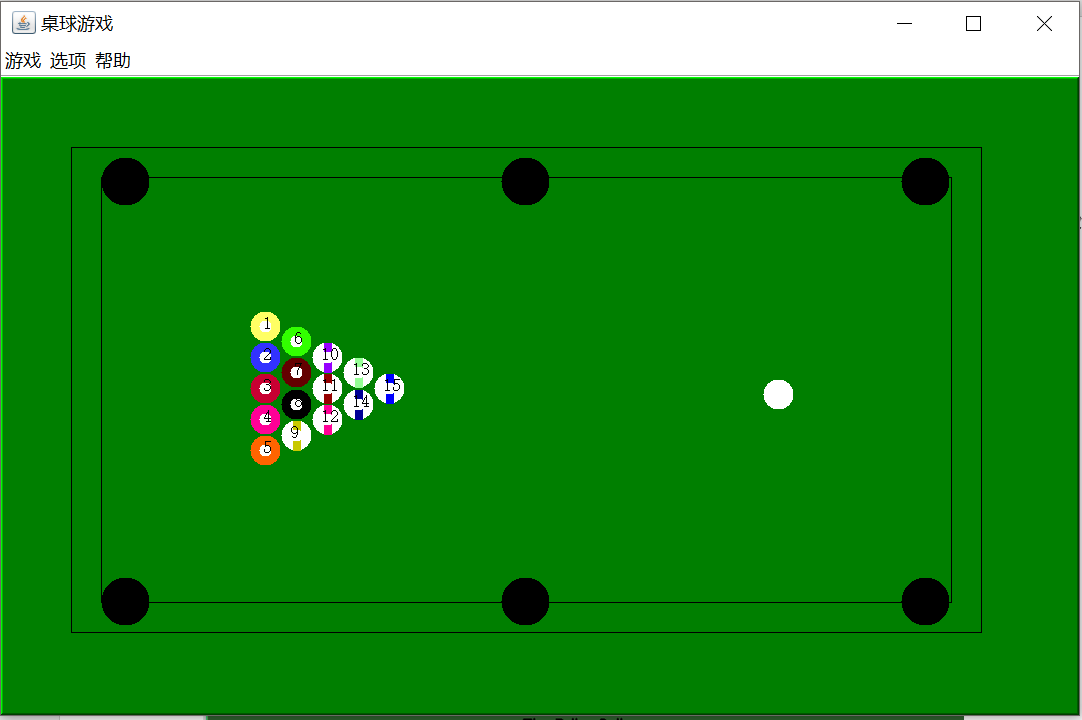
b.dx = kba \* dx - kbb \* dy;

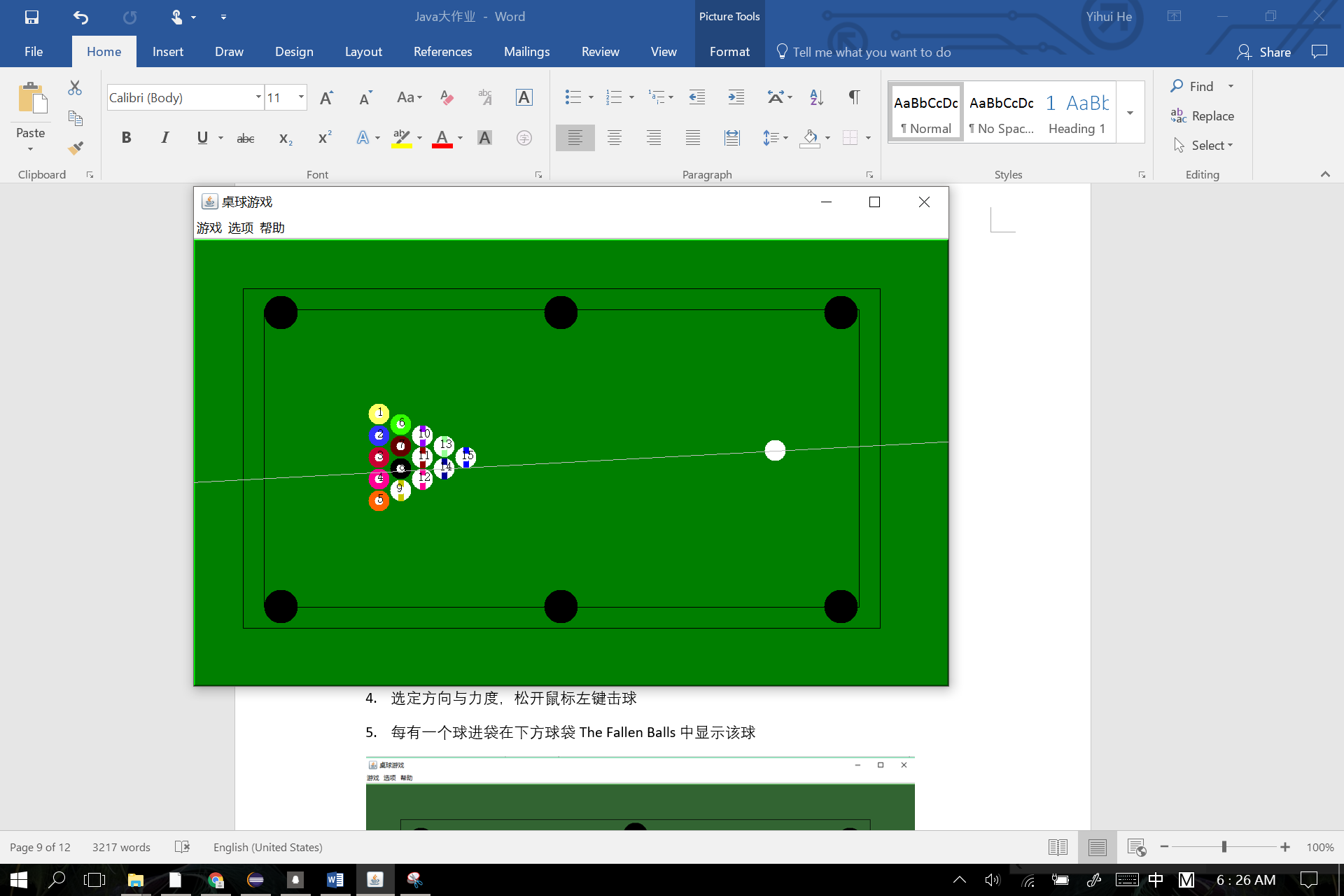
## 实验数据与测试用例准备

首先，我们从<http://www.zapsplat.com/sound-effect-category/snooker-and-pool> 下载了桌球的声音文件。由于下载的是MP3格式，再到<http://www.52ij.com/zhuanhuanqi/yinyue/> 将MP3转换为wav。

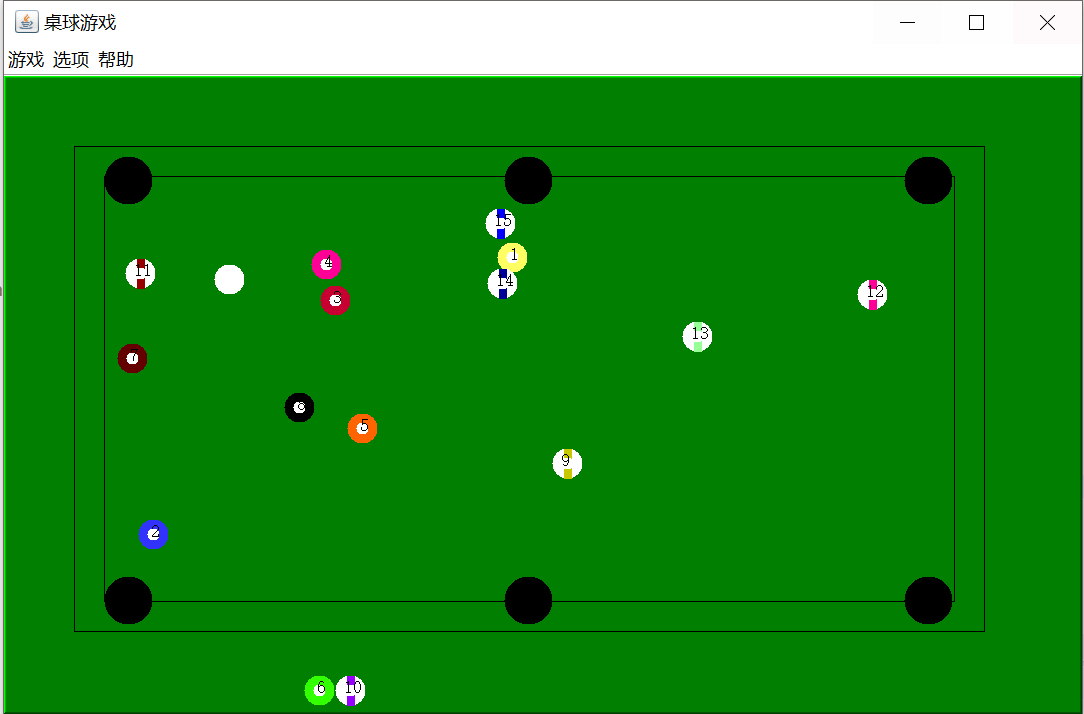
由于我们设计的是桌球游戏，所以我们的测试方式是人人对战。对战中，还是发现程序有一些bug。比如击球力度大的时候，有可能两个球会连在一起。

# 使用说明书

1. 使用eclipse导入桌球游戏项目到工作空间，运行src----->zhuoqiu---->Main.java即开始游戏，得到如下界面。
2. 拖动鼠标选择母球放置位置，单击鼠标左键放下白球。
3. 点击放置好的白球，保持左键不放开，拖动鼠标出现辅助瞄准线及力度线

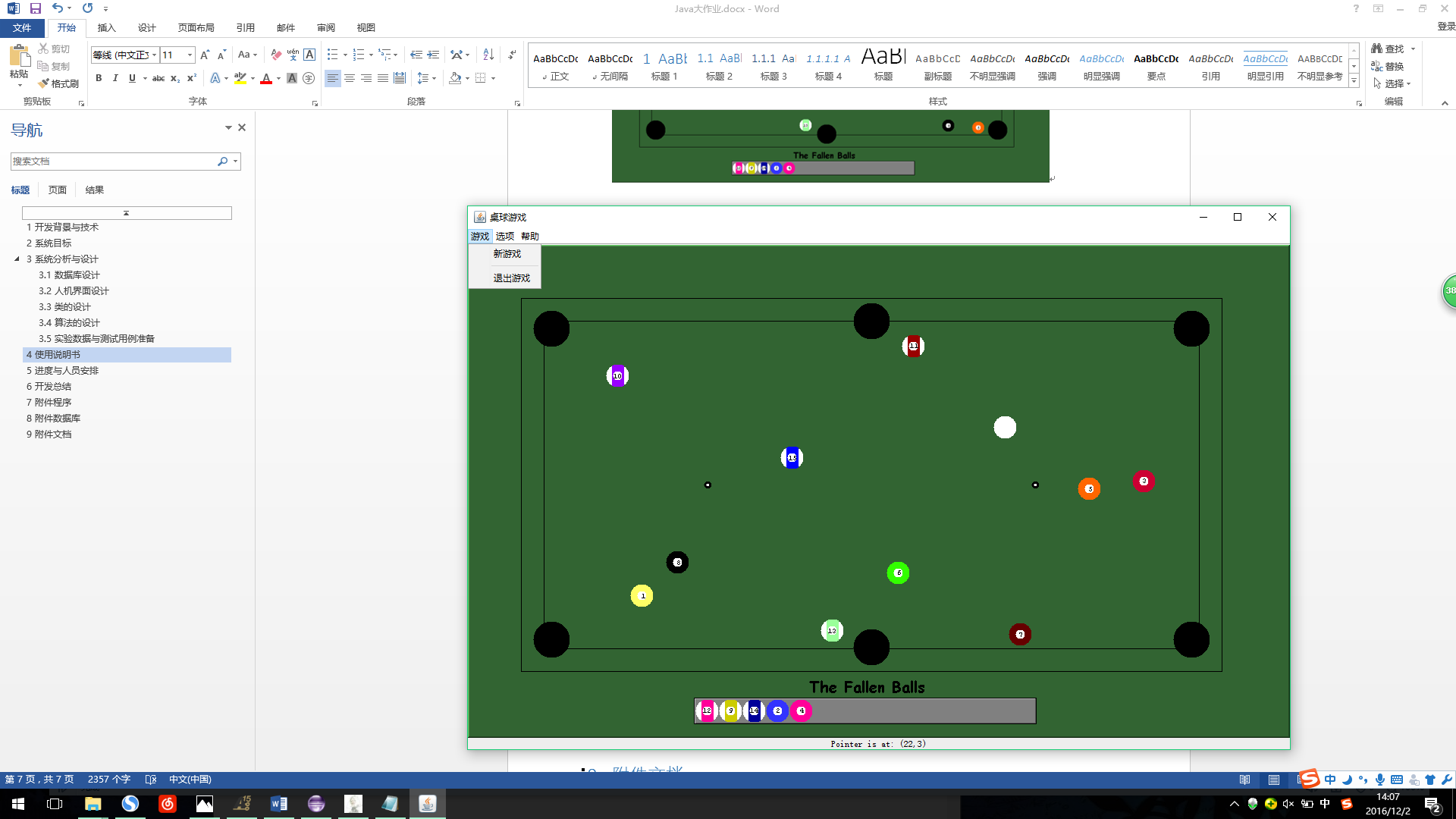


1. 选定方向与力度，松开鼠标左键击球
2. 每有一个球进袋在下方显示该球



1. 若白球进洞则重新摆放白球
2. 所有花球进袋后游戏结束
3. 菜单功能介绍：
4. 游戏菜单

· 新游戏

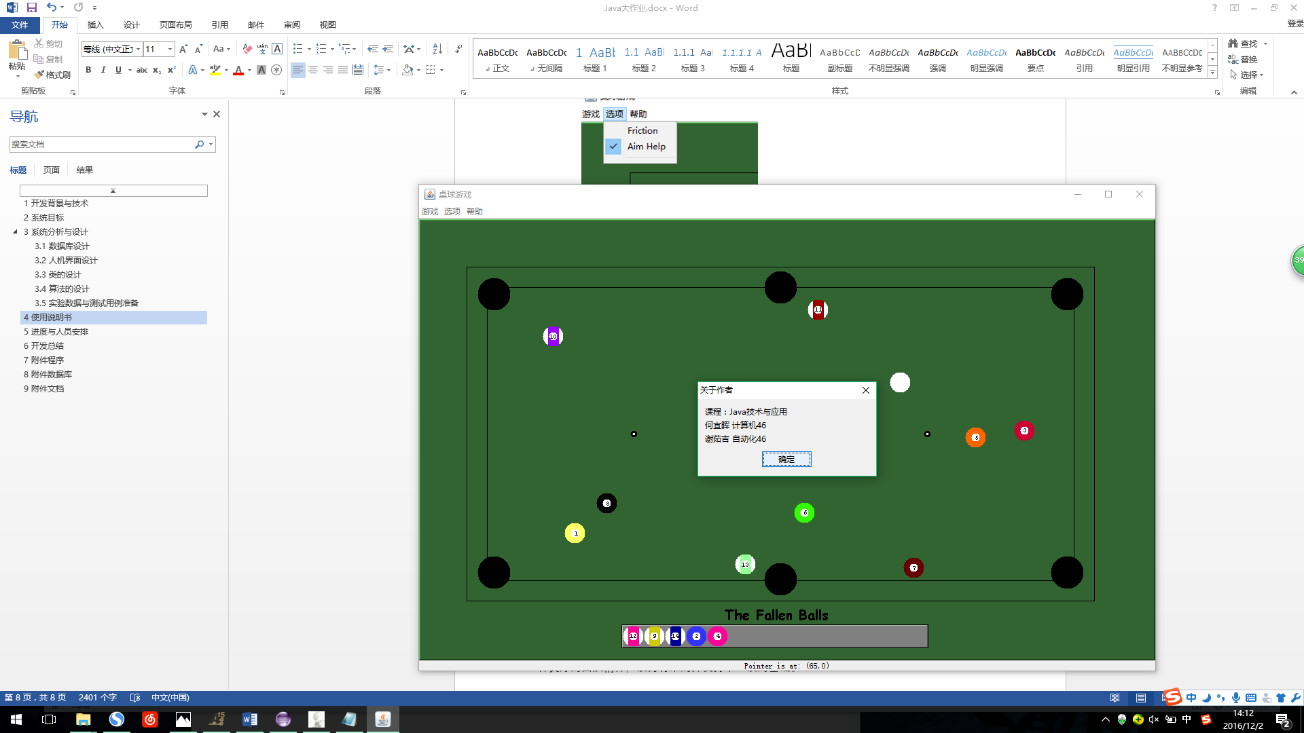
· 退出游戏

1. 选项菜单

· 考虑摩擦 设置有无摩擦力

· 辅助瞄准 设置有误辅助瞄准线

1. 帮助菜单

· 关于

# 进度与人员安排

这次Java大作业，我们利用了Git进行版本控制与团队合作。

**进度控制**，

1. 我们先实现了物理引擎，在一个简单的矩形区域内实现正确的球碰撞的计算。
2. 随后我们基于物理引擎，实现了基本的台球游戏。按照台球规则，加入了球袋，球杆，击球的辅助瞄准。
3. 最后我们加入了声音，以及游戏菜单，方便人机交互。

**人员安排**

* 何宜晖负责写了物理引擎，球袋落球判断，球球碰撞，以及声音类。讨论思路，参与撰写实验报告。
* 谢茹吉负责写了人机交互，台球游戏类，以及主菜单类。讨论思路，参与撰写实验报告。

# 开发总结

在本次课程设计中，我们学到了很多的知识。台球游戏编程需 要一定的物理和数学基础，怎么将学到的理论知识用到实践中去，改变了过去的学习方法和学习态度，同时了解到了完成一个完整项目时团队的重要性。在实际的程序开发中，团队的力量往往比个人力量之和大上很多。对团队开发软件上有了进一步了解，比如一些基本注意点，在团队开发软件的时候不能随便更改类名，方法名，变量及其修饰符，在项目融合时问题会层出不穷，甚至使项目失败，所以规范良好的编程习惯很重要。并且在今后的开发中，我们都将是以团队的形式展开工作的，所以我们应该在现在学习的时候养成一种良好的团队精神，以为将来的开发打下一顶的基础。  
 在这次项目中使我对java面向对象的认识有了更时刻的印象，对象理念有了更深层次的理解，通过和老师的交流和上网了解的一些知识，在游戏开发的周期中编码其实使占很小一部分工作，只有不断具体编程，才能在以后的编码中设计出更加优异和性能更高的代码。

程序中的不足：

1. 摆放白球时，出现球重叠。
2. 多次游戏开局时出现两个白球。
3. 击球力度过大可能会使球重叠。

# 附件程序

见压缩包内源代码。

# 附件数据库

# 附件文档