**《Python程序设计》**

**课程设计报告**

　　　　　　　　　　（2018 —2019 学年第 一 学期）

题 目 弹球小游戏

学生姓名 高 若 云

专业班级 信管16-3

学生学号 311609030304

教师姓名 徐 文 鹏

成 绩**：**

评 语：

教师签名：

日期：

目录

[一、设计目的 2](#_Toc535313478)

[二、设计任务和要求 2](#_Toc535313479)

[三、总体设计 3](#_Toc535313480)

[1.任务分配情况： 3](#_Toc535313481)

[2.程序流程图： 3](#_Toc535313482)

[3.功能模块图： 4](#_Toc535313483)

[四、设计实现 4](#_Toc535313484)

[1.最终实现结果： 4](#_Toc535313485)

[2.实现结果评价： 6](#_Toc535313486)

[五、详细设计 6](#_Toc535313487)

[1.输入输出分析 7](#_Toc535313488)

[2.程序流程图 8](#_Toc535313489)

[3.算法的说明 9](#_Toc535313490)

[六、调试与测试 14](#_Toc535313491)

[1.调试方法 14](#_Toc535313492)

[2.测试结果 15](#_Toc535313493)

[3.测试中遇到的困难 16](#_Toc535313494)

[七、设计总结 16](#_Toc535313495)

# 一、设计目的

课程设计的目的：本课程设计是本专业的一门重要实践性教学环节。在学习了专业基础课和《Python程序设计》课程的基础上，本课程设计旨在加深对Python程序设计的认识，对Python语言及其语言生态有一个进一步的掌握和应用，学会运用Python标准库及外接相关库来解决实际问题的基本能力，培养和提高学生分析问题、解决问题的能力，尤其是提高学生使用Python为开发语言来进行问题描述、交流与思考的能力，为毕业设计和以后的工程实践打下良好的基础。

选择“弹球小游戏”课设的目的：弹球小游戏是一款很有趣味的小游戏，其设计过程有一定的技巧，它既可以加深对python程序设计及其语言生态的认识和进一步掌握，也可以锻炼思维的敏捷性、缜密性和操作的灵活性。其中主要运用到pygame游戏开发工具，pygame是python计算生态里一个开发游戏很好的模块，本次课设通过使用pygame库，致力于设计出一个美观、用户体验良好的“弹球小游戏”。

# 二、设计任务和要求

课程设计的基本要求是：在课程设计的各个阶段严格、规范地完成相关的文档，例如在初期按时完成设计目的、设计要求、总体设计，在后期完成详细设计、调试与测试、设计总结等。要求能完成所选题目的主要功能，程序运行健壮、正确，代码要有详细注释，可读性好；所写文档结构合理、内容完整、叙述清晰。更高要求是：有创意、系统界面美观。

本课程设计具体任务和要求：我们小组的想法是通过pygame库来设计一个美观、用户体验良好的弹球小游戏。在设计中，用户体验第一，要求美观大方，以用户体验为主要出发点，所以在这款游戏里我们选取了优美的图片、音效、以及场景。用户可以通过左右移动挡板反弹小球，小球碰到四周或挡板会改变运动方向，如果挡板没有接到小球，游戏结束，并且返回游戏所用时间，此时用户还可以选择是否重新开始游戏。

# 三、总体设计

## 1.任务分配情况：

“弹球小游戏”代码框架、程序调试和代码整理汇总由小组成员共同完成。

冯孝勇主要负责GameStart函数模块和窗口初始化的编写；

高若云主要负责GameResult函数模块和主函数调用的编写。

## 2.程序流程图：

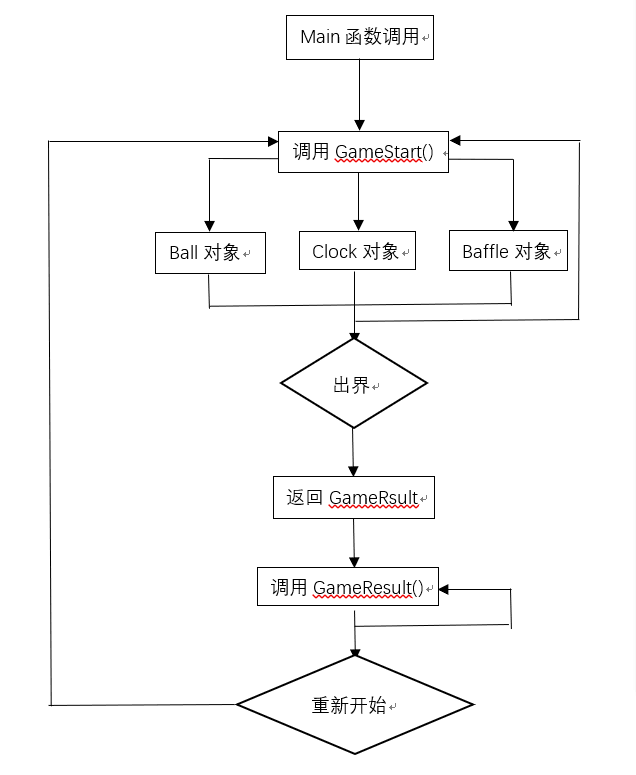


图3-1程序流程图

3.功能模块图：“弹球小游戏”主要包括背景初始化模块、游戏开始模块、游戏结果模块和主函数模块。

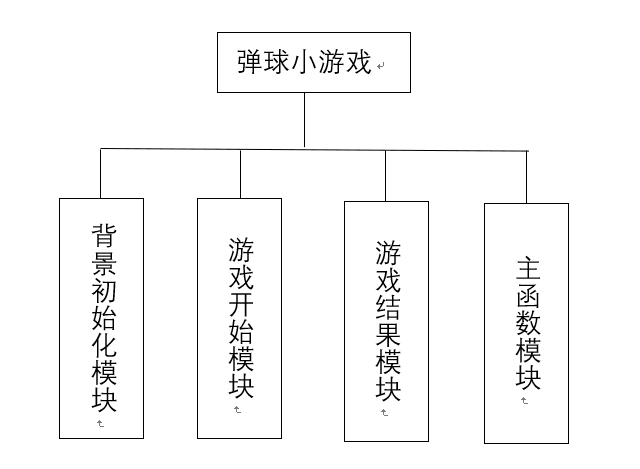


图3-2功能模块图

# 四、设计实现

## 1.最终实现结果：

（1）游戏的启动界面



图4-1启动界面

（2）主要功能运行界面

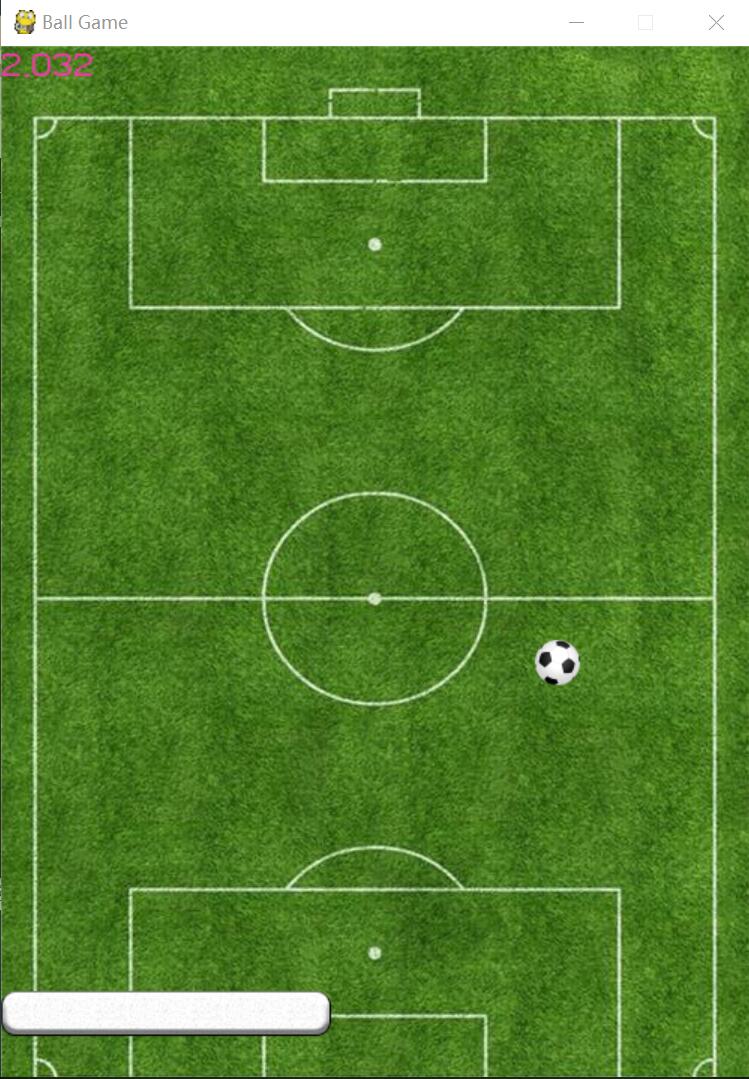


图4-2游戏运行界面

（3）游戏结束界面

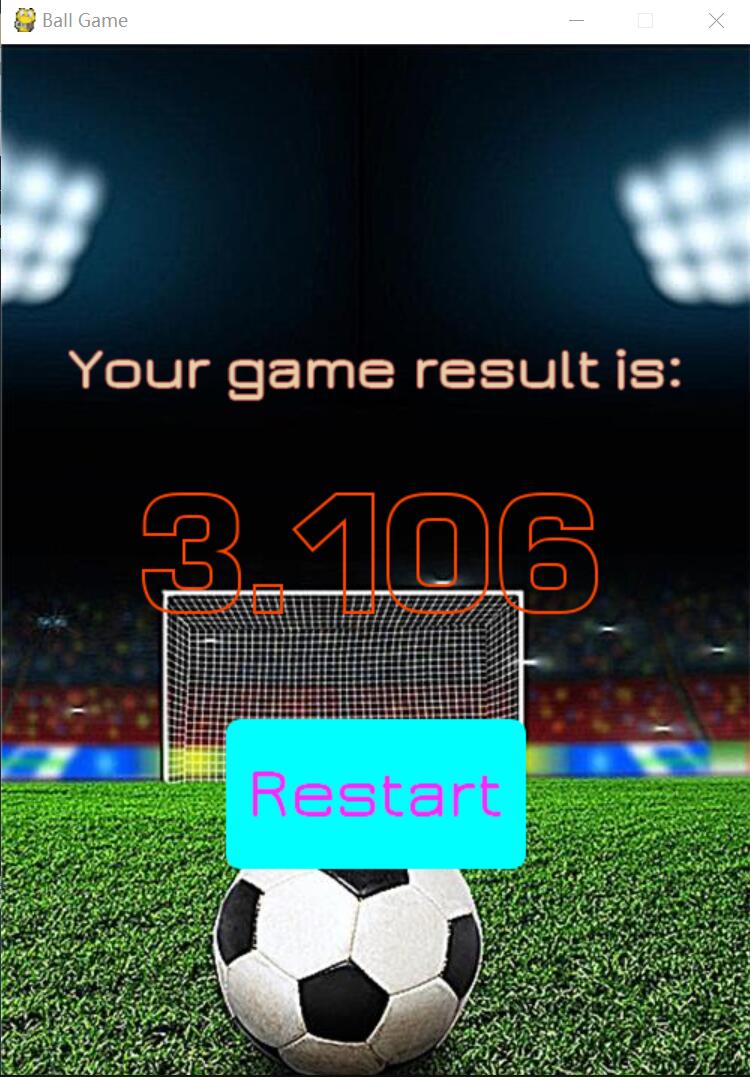


图4-3结束界面

## 2.实现结果评价：

通过设计结果与预期目标比较，我们小组基本完成了最初的想法：通过

pygame库来设计一个美观、用户体验良好的弹球小游戏。实现了用户可以通过左右移动挡板反弹小球，小球碰到四周或挡板会改变运动方向，如果挡板没有接到小球，游戏结束，并且返回游戏所用时间，此时用户还可以选择是否重新开始游戏。但我们的游戏中也存在一些问题，不能实现刚开始的界面，通过点击开始按钮开始游戏，也不能实现在游戏过程中暂停，我相信通过不断的学习，在日后我们小组会完善这款小游戏，增加它的功能，提高它的娱乐价值。

# 五、详细设计

我负责的部分是小球与挡板碰撞的条件判断、游戏结果显示界面、重新开始按钮和主函数模块，通过添加背景和循环控制语句的输出实现游戏结果函数与游戏开始函数的循环调用。

## 1.输入输出分析

（1）输入是一个程序的开始，程序要处理的数据有多种来源，因此形成多种输入方式，内部参数输入是以程序内部的初始化变量为输入，尽管程序看似没有从外部获得输入，但游戏开始函数为程序赋予了执行所需的数据。输入为游戏结束后GameStart函数返回的GameResult参数，既运行时间字符串。

（2）处理是程序对输入数据进行计算产生结果的过程。GameResult函数的处理是将时间字符串整理好格式，如字体、大小、位置等。

Screen.blit(GameResultBackground,(0, 0))

Screen.blit(ResultHint,(45, 200))

RunTimeSurface = GameResultFont.render(GameResult, True,(255, 69, 0))

Screen.blit(RunTimeSurface,(90, 270))

（3）输出是程序展示运算结果的方式，操作系统内部输出指程序将运行结果输出到系统内部变量中。GameResult函数把“Your game result is：”输出到屏幕上，下面显示玩家本次游戏所用的时间，最下面是Restart按钮，实现重新开始游戏的功能。

Screen.blit(ReStartButtonHover, (150, 450))

Screen.blit(ReStartButton, (150, 450))

## 2.程序流程图

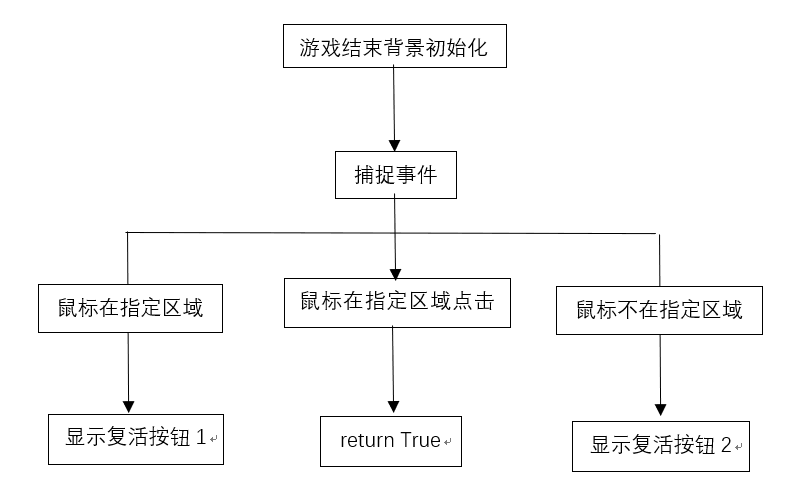


图5-1GameResult函数模块流程图

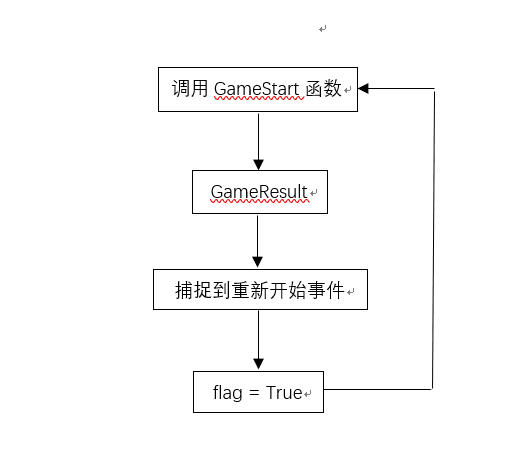


图5-2主函数模块流程图

## 3.算法的说明

（1）定义函数：使用def关键字定义GameResult函数，其中参数为GameResult。

def GameResult(GameResult)

（2）游戏结果显示背景的加载和游戏结果显示字体的加载。

convert函数是将图像数据都转化为Surface对象，每次加载完图像以后就应该做这件事；convert\_alpha相比convert，保留了Alpha 通道信息（可以简单理解为透明的部分），这样光标才可以是不规则的形状。

# 游戏结果背景Surface对象

GameResultBackground=

pygame.image.load('GameResultBackground.png').convert()

# 游戏结果引导

ResultHint = pygame.image.load('ResultFont.png').convert\_alpha()

# 游戏结果Font对象

GameResultFont = pygame.font.Font('EuroBold.ttf', 100)

（3）重新开始按钮图片的加载。

# 重新开始按钮

ReStartButton=pygame.image.load('ReStartButton.png').convert\_alpha()

# 重新开始Hover按钮

ReStartButtonHover=pygame.image.load('ReStartButtonHover.png').convert\_alpha()

（4）while循环：while（）根据括号内部值的真假决定是否进入循环，当条件为True时，循环体重复执行语句块中的语句，无限循环尽管没有输入输出，它也有很好的价值。循环是一个无限循环，在这个主循环里做的事情就是不停地画背景和更新光标位置，虽然背景是不动的，我们还是需要每次都画它，否则鼠标覆盖过的位置就不能恢复正常了，直到用户跳出，才停止循环。

while（True）:

for event in pygame.event.get():

if event.type == QUIT:

exit()

if event.type == pygame.MOUSEBUTTONDOWN

and 150 <= event.pos[0] <= 150 + ReStartButton.get\_width()

and 450 <= event.pos[1] <= 450 + ReStartButton.get\_height():

return True

# 游戏结果背景

Screen.blit(GameResultBackground, (0, 0))

# 游戏结果引导

Screen.blit(ResultHint, (45, 200))

RunTimeSurface = GameResultFont.render(GameResult,True, (255, 69, 0))

Screen.blit(RunTimeSurface, (90, 270))

# 重新开始游戏按钮

MouseX, MouseY = pygame.mouse.get\_pos()

if 150 <= MouseX <= 150 + ReStartButton.get\_width()

and 450 <= MouseY <= 450 + ReStartButton.get\_height():

Screen.blit(ReStartButtonHover, (150, 450))

else:

Screen.blit(ReStartButton, (150, 450))

pygame.display.update()

（5）for循环：根据循环执行次数的确定性，循环可以分为确定次数循环和非确定次数循环，确定次数循环指循环体对循环次数有明确的定义，这类循环在Python中被称为“遍历循环”，其中，循环次数采用遍历结构中的元素个数来体现，具体采用for语句实现。在while循环中通过for循环来实现对事件的获取，通过if语句判断获取的事件，如果接收到退出事件就退出游戏，如果接收

到鼠标点击在重新开始按钮的范围内就return True，调用GameStart函数重新

开始游戏。

for event in pygame.event.get():

if event.type == QUIT:

exit()

if event.type == pygame.MOUSEBUTTONDOWN and 150 <= event.pos[0] <= 150 + ReStartButton.get\_width()

and 450 <= event.pos[1] <= 450 + ReStartButton.get\_height():

return True

（6）把游戏结果背景和游戏结果显示画上去。

Screen.blit（ ，）其中blit是一个重要函数，第一个参数是Surface对象，第二个是左上角的位置，表示把背景图片、背景字体和游戏结果画上去，画完之后要刷新，否则会一片漆黑。每执行一次while循环，都要update一次页面，这样我们才能看到完整的画面。

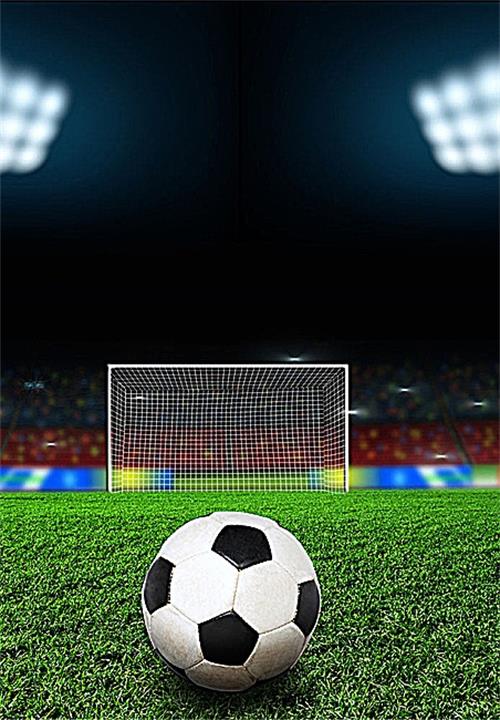


图5-3游戏结果背景

C:\Users\25214\Desktop\新建文件夹\ResultFont.png

图5-4游戏结果提示语

Screen.blit(GameResultBackground, (0, 0))

# 游戏结果引导

Screen.blit(ResultHint, (45, 200))

RunTimeSurface = GameResultFont.render(GameResult, True, (255, 69, 0))

Screen.blit(RunTimeSurface, (90, 270))

# 游戏结果

pygame.display.update()

（7）重新开始游戏按钮

pygame.mouse.get.pos()通过获取鼠标的x坐标和y坐标获得鼠标的具体位置，通过if-else语句判断鼠标的位置是否在Restart按钮的范围内，如果在就显示重新开始按钮图片2，若果不在就显示重新开始按钮图片1。



图5-5重新开始按钮1



图5-6重新开始按钮2

MouseX, MouseY = pygame.mouse.get\_pos()

if 150 <= MouseX <= 150 + ReStartButton.get\_width() and 450 <= MouseY <= 450 + ReStartButton.get\_height():

Screen.blit(ReStartButtonHover, (150, 450))

else:

Screen.blit(ReStartButton, (150, 450))

（8）主函数模块

定义一个flag变量等于True，用一个while无限循环实现GameStart函数和GameResult函数的循环调用，如果GameStart函数返回的结果不是空，则调用GameResult函数显示游戏所用时间，如果鼠标没有点击重新开始按钮，则一直显示游戏结果页面，并不断刷新页面，如果鼠标点击了重新开始按钮，就调用GameStart函数重新开始游戏，从而实现两个函数的循环调用。

if \_\_name\_\_ == '\_\_main\_\_':

flag = True

while flag:

GameResultStr = GameStart()

if GameResultStr != '':

flag = GameResult(GameResultStr)

（9）小球与挡板碰撞条件

从挡板左上角、挡板右上角、挡板左下角、挡板右下角、挡板上表面、挡板下表面、挡板左侧面、挡板右侧面八个方面判断小球与挡板的碰撞条件。

# 挡板左上角

if BallX == BaffleX - Ball.get\_width() and BallY == BaffleY - Ball.get\_height():

BallXSpeed = -BallXSpeed

BallYSpeed = -BallYSpeed

# 挡板左下角

elif BallX == BaffleX - Ball.get\_width() and BallY == BaffleY + Baffle.get\_height():

BallXSpeed = -BallXSpeed

BallYSpeed = -BallYSpeed

# 挡板右上角

elif BallX == BaffleX + Baffle.get\_width() and BallY == BaffleY - Ball.get\_height():

BallXSpeed = -BallXSpeed

BallYSpeed = -BallYSpeed

# 挡板右下角

elif BallX == BaffleX + Baffle.get\_width() and BallY == BaffleY + Baffle.get\_height():

BallXSpeed = -BallXSpeed

BallYSpeed = -BallYSpeed

# 挡板上表面

elif BallX > BaffleX and BallX < BaffleX + Baffle.get\_width() and BallY > BaffleY - Ball.get\_height() and BallY < BaffleY:

BallYSpeed = -BallYSpeed

BallY = BaffleY - Ball.get\_height()

# 挡板下表面

elif BallX > BaffleX and BallX < BaffleX + Baffle.get\_width() and BallY < BaffleY + Baffle.get\_height() and BallY > BaffleY:

BallYSpeed = -BallYSpeed

BallY = BaffleY + Baffle.get\_height()

# 挡板左侧面

elif BallY > BaffleY and BallY < BaffleY + Baffle.get\_height() and BallX > BaffleX - Ball.get\_width() and BallX < BaffleX:BallXSpeed = -BallXSpeed

BallX = BaffleX

# 挡板右侧面

elif BallY > BaffleY and BallY < BaffleY + Baffle.get\_height() and BallX > BaffleX + Baffle.get\_width() - Ball.get\_width() and BallX < BaffleX + Baffle.get\_width():

BallXSpeed = -BallXSpeed

BallX = BaffleX + Baffle.get\_width()

# 六、调试与测试

## 1.调试方法

GameRetult函数调试：通过不断调整背景大小、字体、Restart按钮位置使整个页面协调，工整。

主函数模块调试：不断测试循环调用的条件与方法实现无限循环，使程序能完好地运行。

球与挡板碰撞条件调试：通过不断测试球与挡板不同位置的碰撞，最后得出八个方面，实现小球在碰撞后的反弹，使小球不断改变方向，最后判断游戏结束条件，既小球未能与挡板碰撞掉落，此时游戏不能在继续，游戏结束，返回游戏运行时间。

## 2.测试结果

GameRrsult函数和主函数测试结果：



图6-1

小球与挡板碰撞运行结果：

判断游戏结束的条件：

if BallY > 720 - 45:

GameResult = RunTimeStr

return GameResult

# 刷新显示

pygame.display.update()

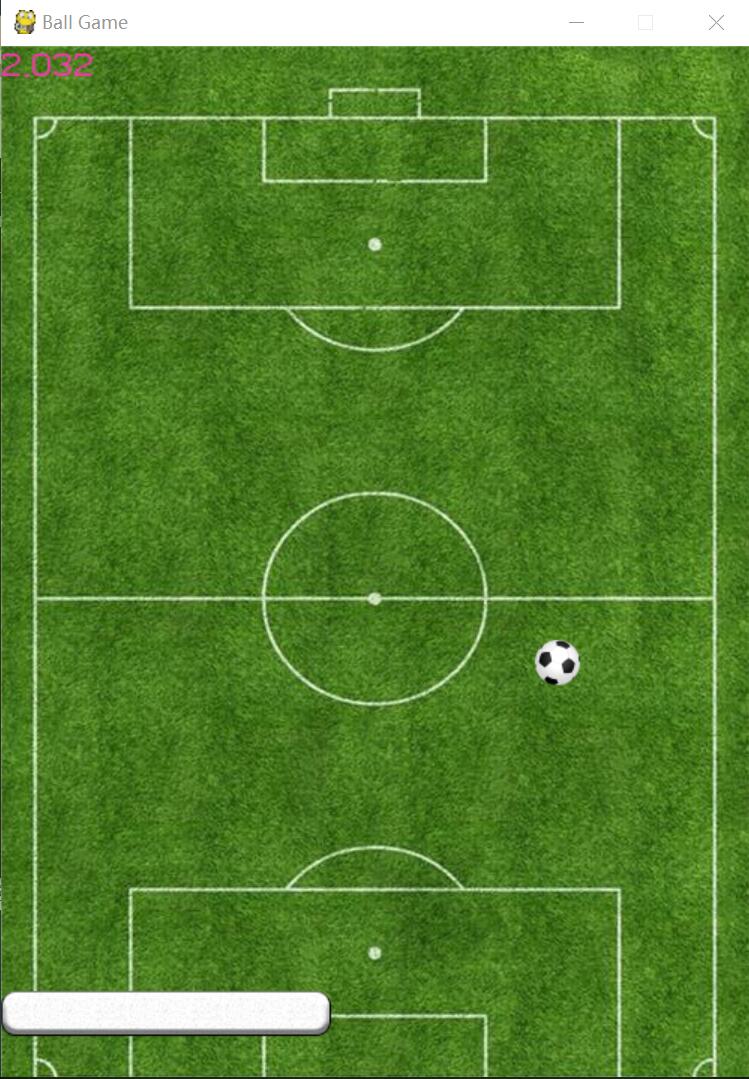


图6-2

3.测试中遇到的困难：在测试过程中刚开始找不准重新开始按钮键的位置，通过一次次的调试找到了合适的位置，球与挡板碰撞条件的判断刚开始写不出准确的判断条件，最后通过画图数学分析找出的所有条件。

# 七、设计总结

首先我要由衷的感谢老师提供给我这样一个锻炼自己的机会，经过这一周的学习，我的收获很大。我对pygame游戏开发库有了一个初步的了解，可以编写一些简单的小游戏。回顾这段时间的课程设计，从选题到开始实现，从理论到实践，不仅巩固了以前所学过的知识，而且学到了很多在书本上所没有学到过的知识。通过这次课程设计使我更加懂得并亲身体会到了理论与实际相结合的重要性，只有理论知识是远远不够的，只有把所学的理论知识与实践相结合起来，从实践中得出结论，才能提高自己的实际动手能力和独立思考的能力。

在设计的过程中遇到很多问题，但都通过小组讨论和求助同学解决了，在设计的过程中发现了自己的很多不足之处，发现自己对之前所学过的知识理解得下够深刻，掌握得下够牢固，有待加强。本次课程设计小组的游戏也有很多不许之处，游戏碰撞音效我们没能添加成功，还有挡板的大小、位置设计不太合理，游戏过程中也没能实现暂停功能，我相信这些不足与缺陷随着我们以后的不断学习最终会实现。