

**《Python程序设计》**

**课程设计报告**

　　　　　　　（2018 —2019 学年第 一 学期）

题 目 《吃东西小游戏》

学生姓名 范 坤 鹏

专业班级 信管16-3

学生学号 311609030315

教师姓名 徐 文 鹏

成 绩**：**

评 语：

教师签名：

日期：

目录

[一. 设计目的 3](#_Toc535088070)

[二．设计任务和要求 3](#_Toc535088071)

[三．总体设计 4](#_Toc535088072)

[四．设计实现 6](#_Toc535088073)

[（1）最终实现结果： 6](#_Toc535088074)

[（2）实现结果评价 7](#_Toc535088075)

[五．详细设计 7](#_Toc535088076)

[（1）pygame事件 7](#_Toc535088077)

[（2） 鼠标事件 8](#_Toc535088078)

[（3）轮询键盘 8](#_Toc535088079)

[（4）游戏结束的处理 9](#_Toc535088080)

[（5）游戏等级设置和难度设置 9](#_Toc535088081)

[（6）游戏过程中的介绍 10](#_Toc535088082)

[六．调试与测试 12](#_Toc535088083)

[（1）异常处理 12](#_Toc535088084)

[（2）调试 12](#_Toc535088085)

[七.设计总结 12](#_Toc535088086)

# 设计目的

课程设计教学目的：

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。

本课程设计具体目的：

（1）为全面了解Python技术历史、现状与发展趋势的基础上，系统掌握Python基本概念、编程思想以及程序设计技术，具备熟练的Python编程技能和面向对象软件设计技术思想。

（2）通过总体框架到页面设计再到代码实现，了解Python设计的一般步骤，并熟练掌握Python的设计思想。

（3）本课题主要训练逻辑思维和Python语法，对代码编写由大到小，由总体到模块，掌握其基本逻辑，对以后的设计可以做到举一反三，进而形成逻辑思维。

# 二．设计任务和要求

课程设计教学任务和要求：

本课程设计主要任务是以Python为开发语言完成一个100~300行左右规模的程序项目开发。设计参考题目见附件1。

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，例如总体方案报告，详细设计报告、功能说明、数据结构说明、算法说明、程序设计框图、图例和源程序等。要求所写文档结构合理、内容完整、叙述清晰。程序源码要有详细注释，可读性好。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：

（1）页面完美，逻辑清晰。

（2）代码编写在可读性高的情况下尽可能的简洁。

（3）可选用PyCharm，Python IDLE等开发平台以提高开发效率，通过资料查阅和学习尽可能掌握一种或多种集成开发环境。

（4）共同完成代码的衔接，游戏总体框架和逻辑的实现。分工编写课程设计报告，报告总共包括六大部分：设计目的、设计要求、总体设计、详细设计、调试与测试、设计总结。

# 三．总体设计

反复调试测试

反复调试测试

整体调试测试

优化

设计游戏逻辑

设计游戏整体框架

设计游戏构型

定义变量规范

Score>600

触碰炸弹？

Pic=pic2

Lives--

Round 3

vel\_y=0.8

加入炸弹

Score>900

Round 4

vel\_y=1.0

Round 5

vel\_y=1.2

Round 6

vel\_y=1.2

Score>1200

Score>=15000

游戏开始

Lives=10

Score=0

Round 1

vel\_y=0.4

Round 2

vel\_y=0.6

Score>300

Lives=0?

end

有鱼没接？

Lives--

No

Yes

Yes

Yes

No

Yes

Yes

Yes

Yes

No Yes

Yes

Yes

我负责游戏逻辑的实现。

Vel\_y是游戏中鱼下落的速度，游戏中每当接到了炸弹，猫的表情会变成哭脸，同时生命值减一，遗漏鱼时，生命值也会减一，当生命值为0时，游戏结束。

# 四．设计实现

（1）最终实现结果：

游戏的启动页面：

图1 游戏启动界面

游戏的运行界面：

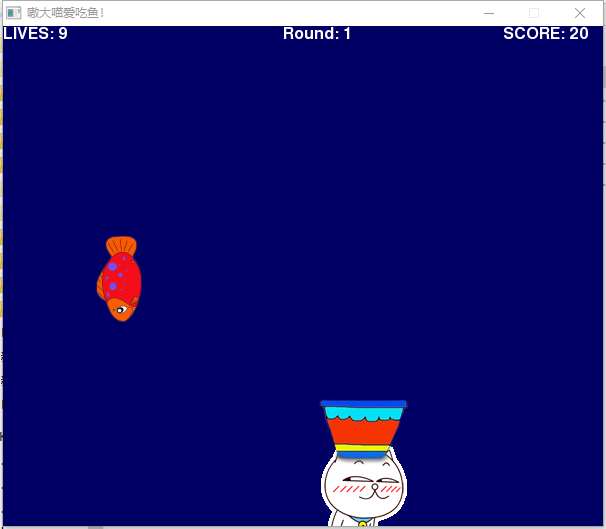


图2 游戏运行界面

当Round>3时，会插入炸弹：

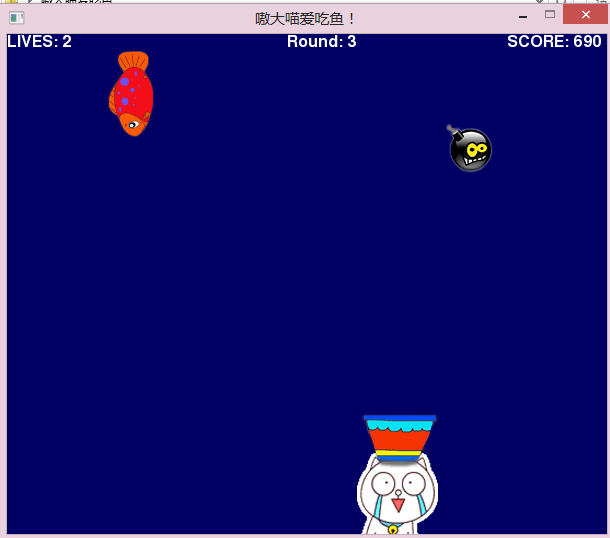


图3 炸弹界面

（2）实现结果评价

良好

总体上来说，游戏的运行基本满足了当初设计时的构想，满足了设计要求。

# 五．详细设计

游戏具体逻辑的实现：

（1）pygame事件

pygame事件可以处理游戏中的各种事情，pygame中的事件处理是放在一个实时的循环中来完成的。将代码都放在一个while True的循环中，但是这样会造成死循环，所以在里面加一句sys.exit()来退出。

for event in pygame.event.get()

上面的代码将会创建当前等待处理的事件的一个列表，然后使用for循环来遍历里面的事件。这样，我们将会根据事件产生的顺序依次地进行不同的操作。常见的事件是按键按下，按键释放以及鼠标移动。通常需要最先处理QUIT事件（在用户关闭窗口的时候会产生该事件。）

while True:

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

sys.exit()

（2） 鼠标事件

pygame支持一些鼠标事件，他们包括：

MOUSEMOTION,MOUSEBUTTONUP,MOUSEBUTTONDOWN.

在MOUSEMOTION中包含了一些属性：event.pos,event.rel,event.buttons

Elif event.type == MOUSEMOTION:

mouse\_x,mouse\_y = event.pos

move\_x,move\_y = event.rel

MOUSEBUTTONUP里面的属性：

elif event.type == MOUSEBUTTONUP:

if game\_over:

game\_over = False

lives = 10

score = 0

Round =1

vel\_y=0.4

mine=0

flag=0

pic=cat

bomb\_y = -50

（3）轮询键盘

在pygame中，使用pygame.key.get\_pressed()来轮询键盘接口。这个方法会返回布尔值的一个列表，其中每个键一个标志。使用键常量值来匹配按键，这样的好处就是不必遍历事件系统就可以检测多个键的按下。

keys = pygame.key.get\_pressed()

if keys[K\_ESCAPE]:

sys.exit()

（4）游戏结束的处理

if game\_over:

screen.blit(init,(60, 60))

print\_text(font3, 200, 400,"Clicked To Play!")

print\_text(font2, 310, 480,"Copyright@2015 developed by xiaoxiami")

图4 游戏结束页面

（5）游戏等级设置和难度设置

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 等级 | 1 | 2 | 3 | 4 | 5 | 6 |
| 分数 | <300 | 300600 | 600900 | 9001200 | 12001500 | >1500 |
| 下落速度 | 0.4 | 0.6 | 0.8 | 1.0 | 1.2 | 1.2 |

else:

#Round setting

if score >300 and score <600:

Round=2

elif score >600 and score <900:

Round =3

elif score >900 and score <1200:

Round=4

elif score >1200 and score <1500:

Round =5

elif score >=1500:

Round =6

#draw the Round

print\_text(font1, 280, 0, "Round: " + str(Round))

#speed setting

if Round ==1:

vel\_y=0.4

elif Round ==2:

vel\_y=0.6

elif Round ==3:

vel\_y=0.8

elif Round ==4:

vel\_y=1.0

elif Round ==5:

vel\_y=1.2

（6）游戏过程中的介绍

如果错过鱼的话，就重置鱼的位置，给它一个随机的x值，然后生命值减一：

if bomb\_y > 500:

bomb\_x = random.randint(0, 500)

bomb\_y = -50

lives -= 1

if lives == 0:

game\_over = True

简单碰撞检测函数，查看是否接住鱼：

elif bomb\_y > pos\_y:

if bomb\_x > pos\_x-10 and bomb\_x < pos\_x + 70:

score += 10

bomb\_x = random.randint(0, 500)

bomb\_y = -50

同理炸弹的检测和这个是类似的：

if mine\_y > 500:

mine\_x = random.randint(0, 500)

mine\_y = -50

为了控制猫的坐标不超出屏幕范围,加入了以下代码：

pos\_x = mouse\_x

if pos\_x < 0:

pos\_x = 0

elif pos\_x > 510:

pos\_x = 500

猫在接到炸弹。或者生命值小于5的时候，会变成哭脸，因此我们还需要加载一张哭脸的位图，然后在程序中添加一些相应的逻辑代码：

加载猫的哭脸位图：

cat2=pygame.image.load("aodamiao\_3.png")

当接到炸弹的时候，猫变成哭脸：

elif mine\_y > pos\_y:

if mine\_x > pos\_x and mine\_x < pos\_x + 40:

mine\_x = random.randint(0, 500)

mine\_y = -50

lives-=1

pic=cat2

if lives == 0:

game\_over = True

当猫的生命值小于5时，猫变成哭脸：

if lives<5:

pic=cat2

# 六．调试与测试

（1）异常处理

**try...except...finally...**

**用于判断猫的位置是否超出了游戏窗口：**

try:

print("开始执行try...")

mouse\_x>0

mouse\_x<510

except ZeroDivisionError as e:

print(“猫超出范围”)

finally:

print("执行finally...")

print("END...")

（2）调试

Print

通过print 输出可能出错的变量，可以通过错误信息和print 输出的内容来进行修改bug，print在测试完成后还需要删除

pdb

启动Python的单步调试器pdb，程序一行行执行。可以采用pdb.set\_trace()在代码中设置断点

# 七.设计总结

这次做论文使我受益良多。一方面，通过这次实验的完成，我对Python有了更深刻的理解，能够更加熟练的运用这门语言，更加理解到了编程的魅力。而另一方面，使我明白了认真做事的重要性。使我体会到完成一件事的喜悦。以后面对困难的时候，都能有直面困难并克服它的决心。本设计能够顺利的完成，也感谢老师的辅导和同学们的互相帮助学习，使我能够很好的掌握和运用专业知识，并在设计中得以体现。在以后的学习生活中我一定会能有更好的发挥，克服自己的缺点，不断改进，不断成长！