

阿里云天池牛年读书会

# Python游戏趣味编程

分享嘉宾：童晶  
浙大计算机博士、河海大学副教授

# 天池读书会

TIANCHI 天池

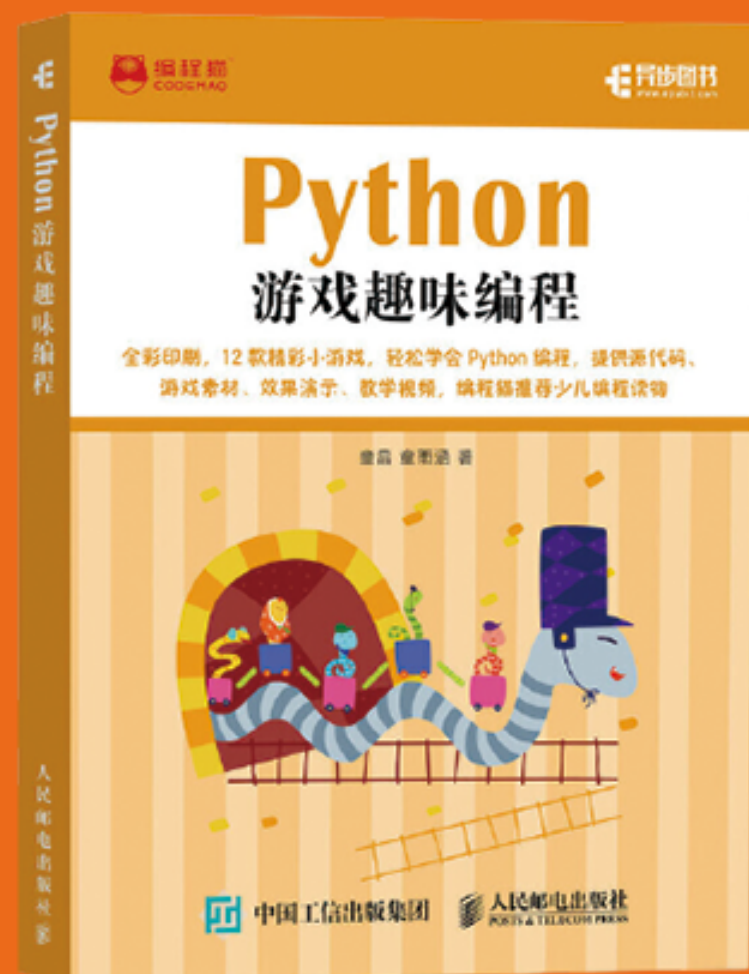
异步社区  
人民邮电出版社

## 《Python游戏趣味编程》

针对Python教学枯燥乏味的问题，通过趣味游戏开发零基础学习Python编程，上手容易、有成就感。

直播嘉宾：童晶 浙大博士、河海大学副教授

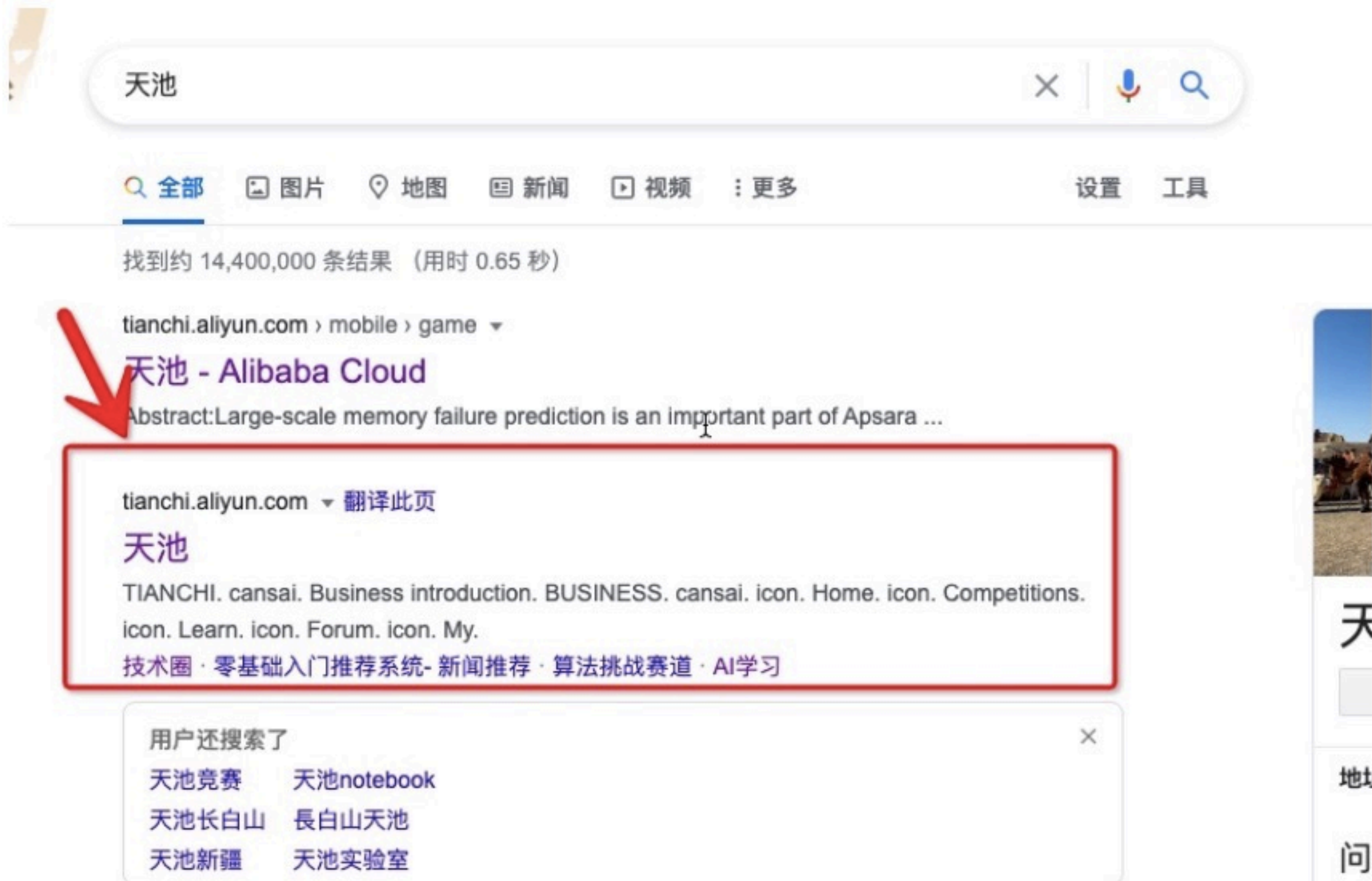
直播时间：6月23日20:00



扫码领取读书会配套学习资源



1) 首先需要进入天池官网，大家打开浏览器，搜索 天池，找到 tianchi.aliyun.com即可访问进入天池官



网；

2) 在天池官网，将鼠标移到 天池学习，即可出现下拉列表，点击 天池读书会，即可进入天池读书会的页面。



3) 在天池读书会页面，你可以对对应的读书会图书进行提问，优秀的提问还有机会获得赠书，还可以点击配套的训练营或者课程资源进入学习，还有点击实践代码获取读书会的项目实践的代码，跟着我一起进行项目实践和代码学习，同时还有很多其他的读书会，大家也可以观看举办过的读书会的回放，或者预约还没开始的读书会。



**童晶** 本书作者、浙大博士、河海大学副教授

直播主题 《Python游戏趣味编程》

直播时间 2021年6月23日 20:00

学习资料 Python训练营

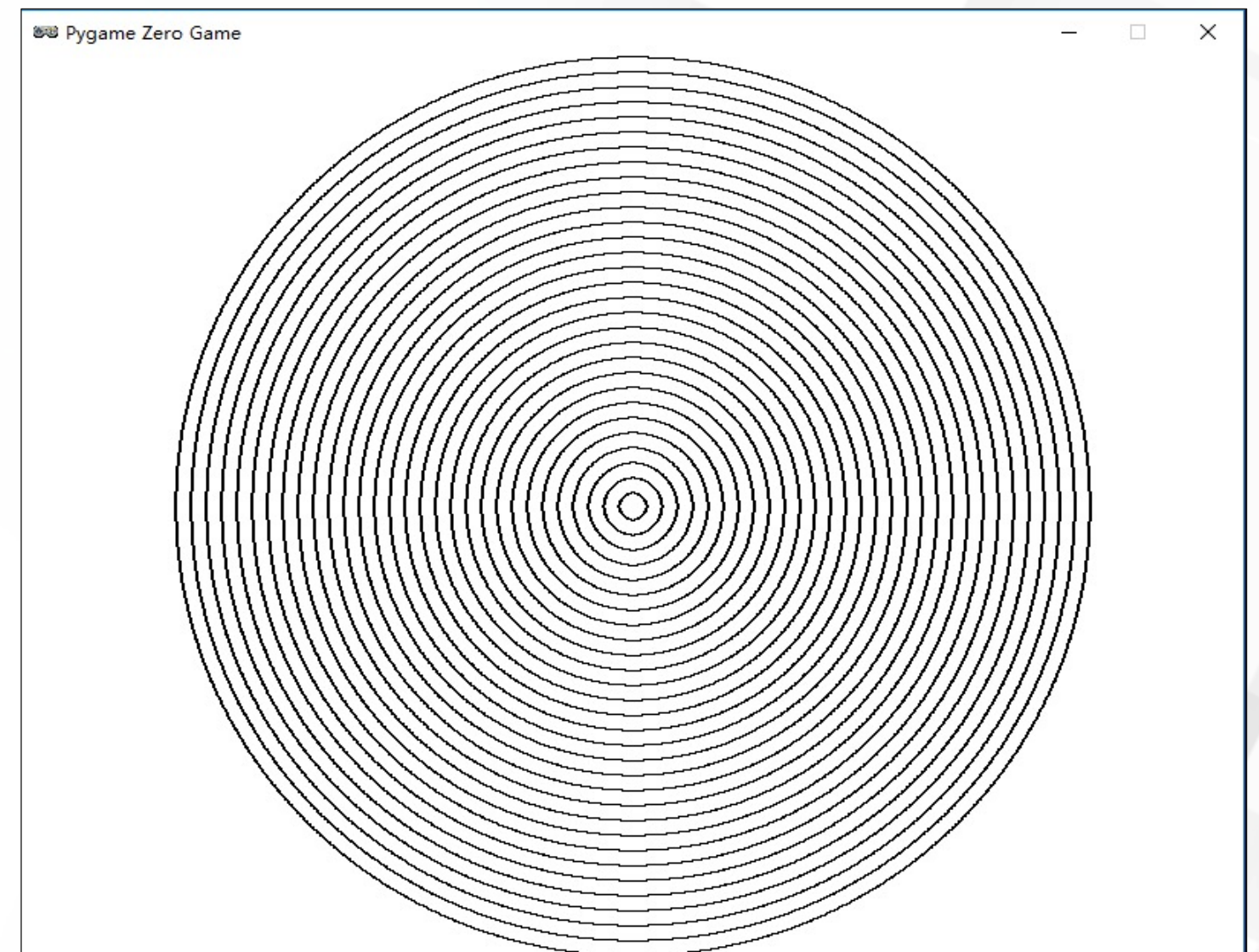
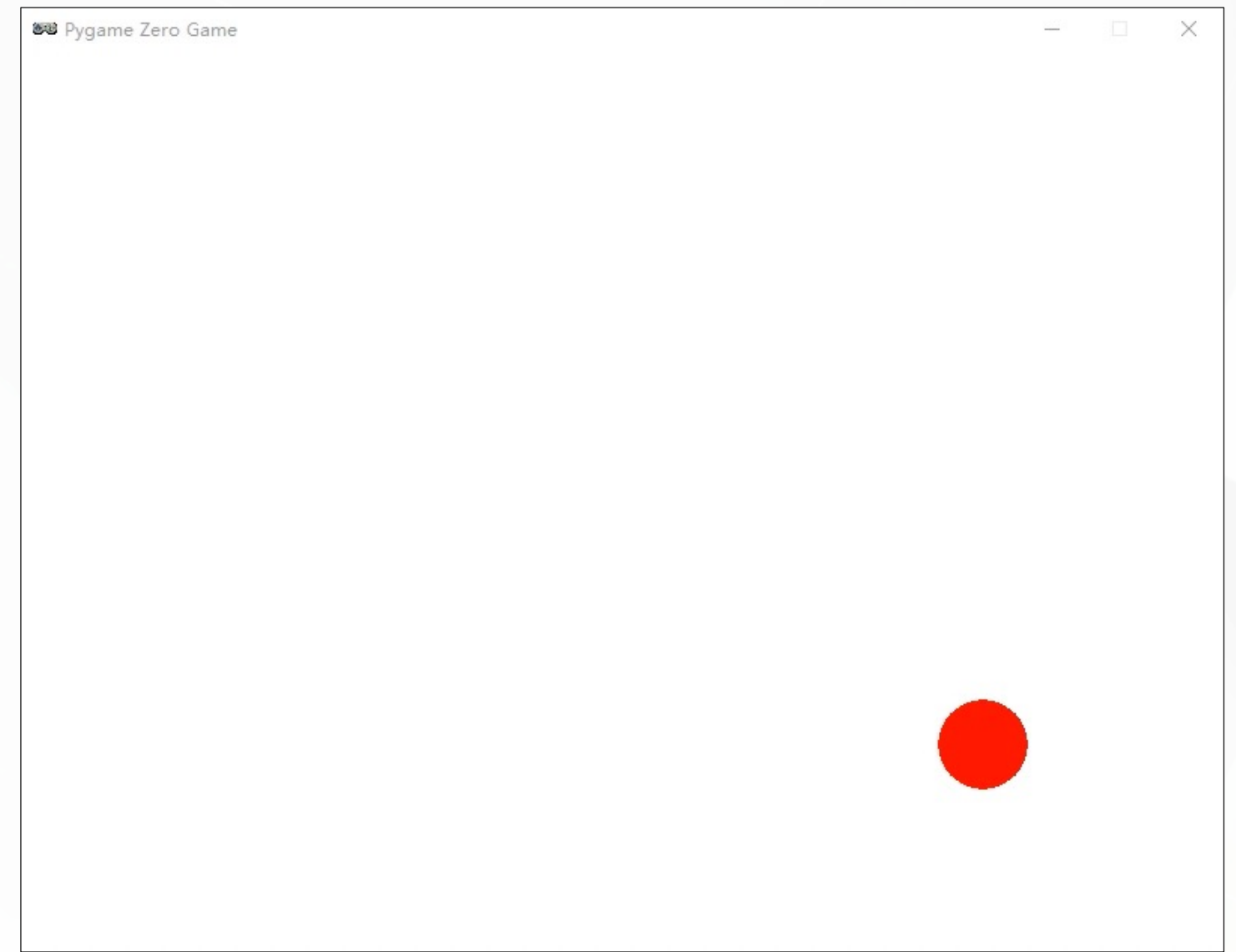
实践项目 待定



[🗨️ 提问](#) | [📖 学习训练营](#) | [🛒 购买地址](#) | [📄 PPT下载](#) | [👉 实践代码](#) | [🕒 预约直播](#)

# 目录

1. 分享嘉宾简介
2. 图书写作背景
3. 项目实践1：弹跳的小球
4. 项目实践2：绘制同心圆
5. 图书简介
6. Q&A 答疑



## 分享嘉宾简介



### 童晶

浙江大学计算机专业博士

河海大学计算机系副教授、硕士生导师

曾获中国发明创业成果奖一等奖、浙江省自然科学二等奖

著有《Python游戏趣味编程》、《Python趣味创意编程》、《C和C++游戏趣味编程》、《C语言课程设计与游戏开发实践教程》等图书

## 图书写作背景

- 教了16年编程基础课，近年来主要教授程序设计基础课程
- 然而.....





## 图书写作背景

### • 传统编程教学方法的问题

- 讲解偏语法，实例偏数学算法、趣味性不强
- 学生讨厌编程，不会真正写程序

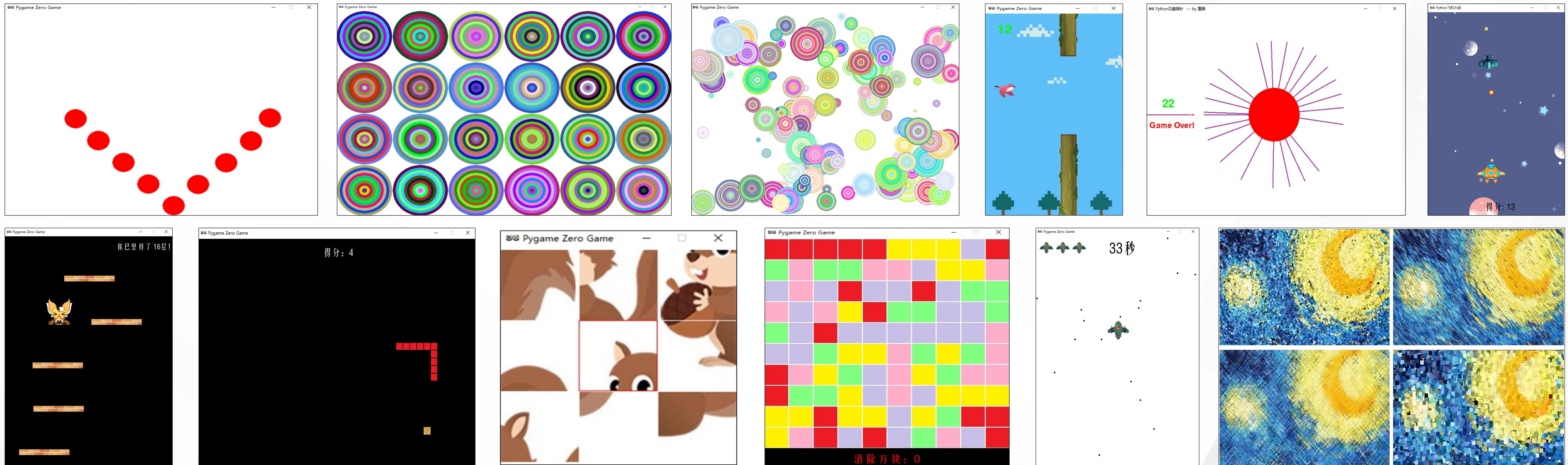
如  $a=2, b=3, c=-1$ ，执行  $++a || ++b || ++c$ ；后， $a$  的值为\_\_\_\_\_， $b$  的值为\_\_\_\_\_， $c$  的值为\_\_\_\_\_，表达式的值为\_\_\_\_\_

```
def isPrime(num):  
    for i in range(2, num):  
        if num % i == 0:  
            return False  
    else:  
        return True
```

# 图书写作背景



- 通过趣味游戏开发，学习Python编程
- 语法知识穿插在12个案例开发过程中讲解，上手容易，有成就感，可拓展性强



# 实战演示、互动交流

阿里云 | TIANCHI 天池 | 异步社区 人民邮电出版社

## 天池读书会

### Python游戏趣味编程

分享嘉宾：童晶 浙大博士、河海大学副教授

直播时间：6月23日 20:00

直播通道：@天池读书会  
@B站达摩院扫地僧

扫码预约观看直播

针对Python教学枯燥乏味的问题，通过趣味游戏开发零基础学习Python编程，上手容易、有成就感。

- 01 如何通过游戏学习Python编程
- 02 利用Python编写一个弹跳的小球程序
- 03 利用Python编写一个绘制同心圆程序

浙大博士、河海大学副教授教你如何从游戏里零基础入门Python编程

大家可以使用手机扫左侧海报二维码，或者电脑访问下方地址进入天池读书会页面，点击今天读书会中的 **实践代码** 和我一起进行项目实践学习，天池为大家准备好了代码和运行环境，非常方便。

<https://tianchi.aliyun.com/specials/promotion/activity/bookclub>



**童晶** 本书作者、浙大博士、河海大学副教授

直播主题 《Python游戏趣味编程》

直播时间 2021年6月23日 20:00

学习资料 Python训练营

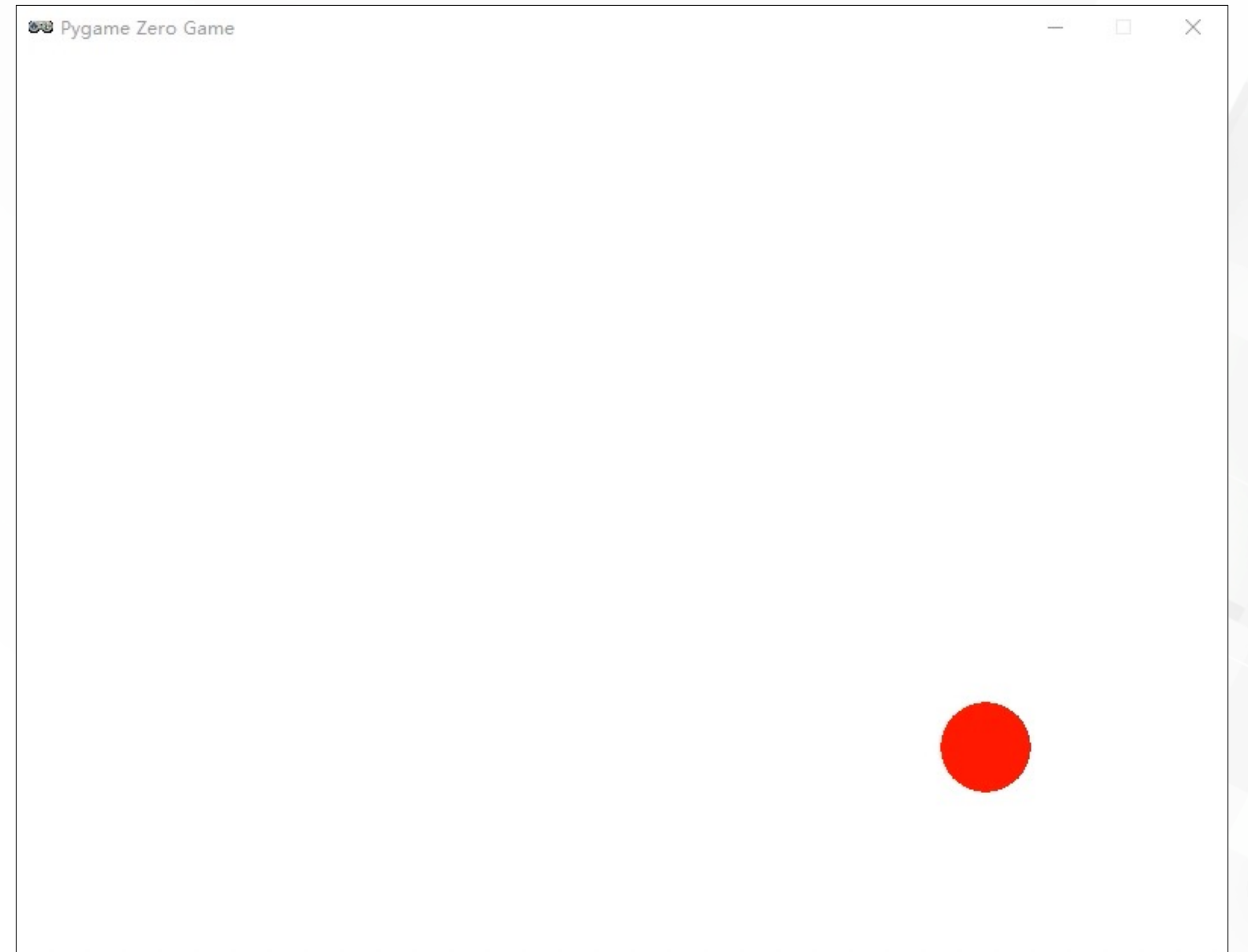
实践项目 待定



[🗨️ 提问](#) | 
 [📖 学习训练营](#) | 
 [🛒 购买地址](#) | 
 [📄 PPT下载](#) | 
 [👉 实践代码](#) | 
 [🕒 预约直播](#)

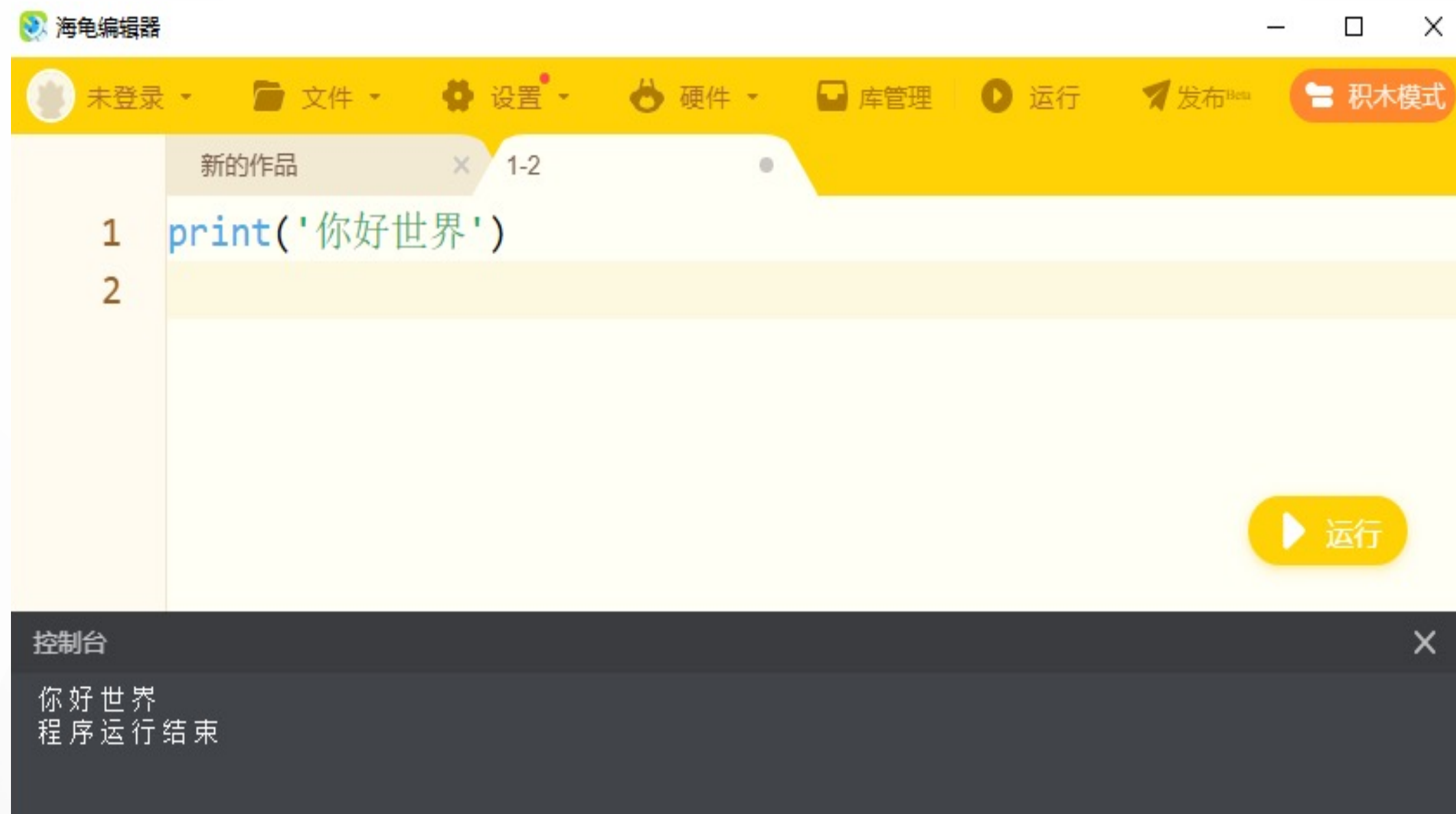
## 项目实践1：弹跳的小球

- 1 导入游戏开发库
- 2 显示一个静止小球
- 3 设置小球与背景颜色
- 4 显示多个静止小球
- 5 利用变量修改多个小球的参数
- 6 逐渐变大的小球
- 7 小球逐渐下落
- 8 利用if语句实现小球重复下落
- 9 小球上下反弹
- 10 斜着弹跳的小球



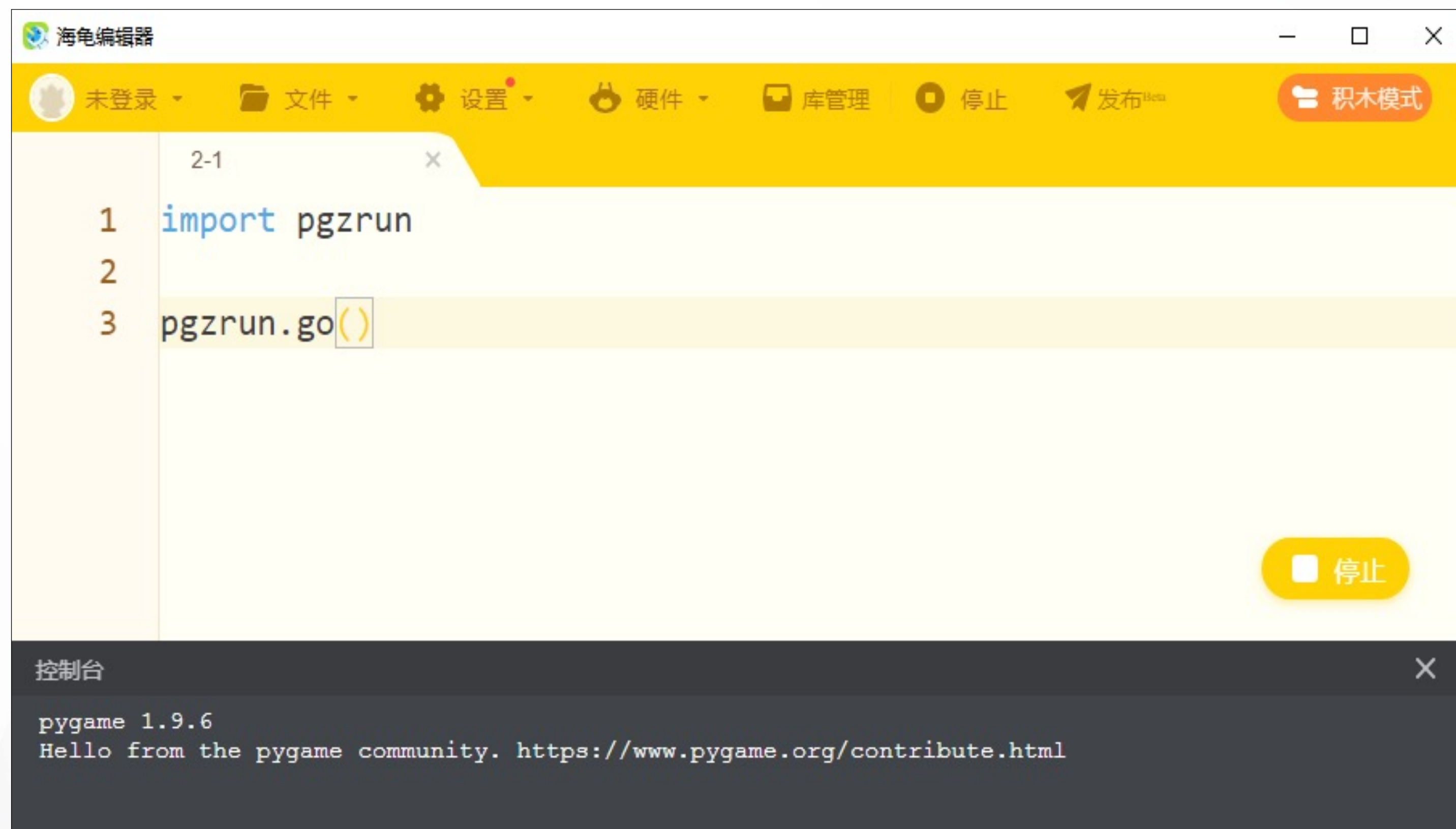
## 项目实践1：弹跳的小球

- 要让计算机读懂Python程序，需安装集成开发环境
  - 编程猫海龟编辑器 (<https://python.codemao.cn/>)
  - Mu编辑器 (<https://codewith.mu/en/download>)



## 项目实践1：弹跳的小球

```
import pgzrun  
pgzrun.go()
```

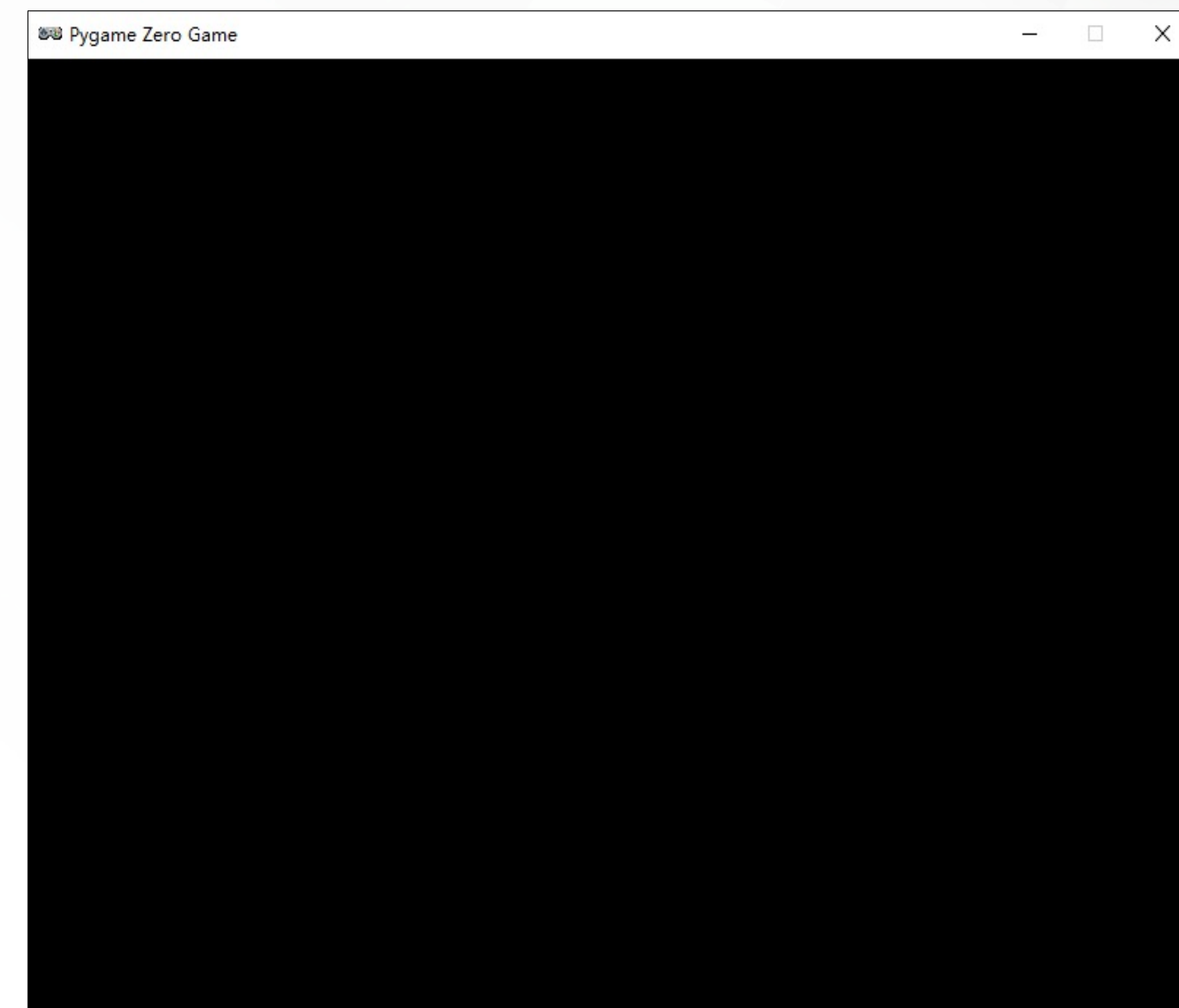


The screenshot shows the SeaTurtle IDE interface. The main editor area contains the following code:

```
1 import pgzrun  
2  
3 pgzrun.go()
```

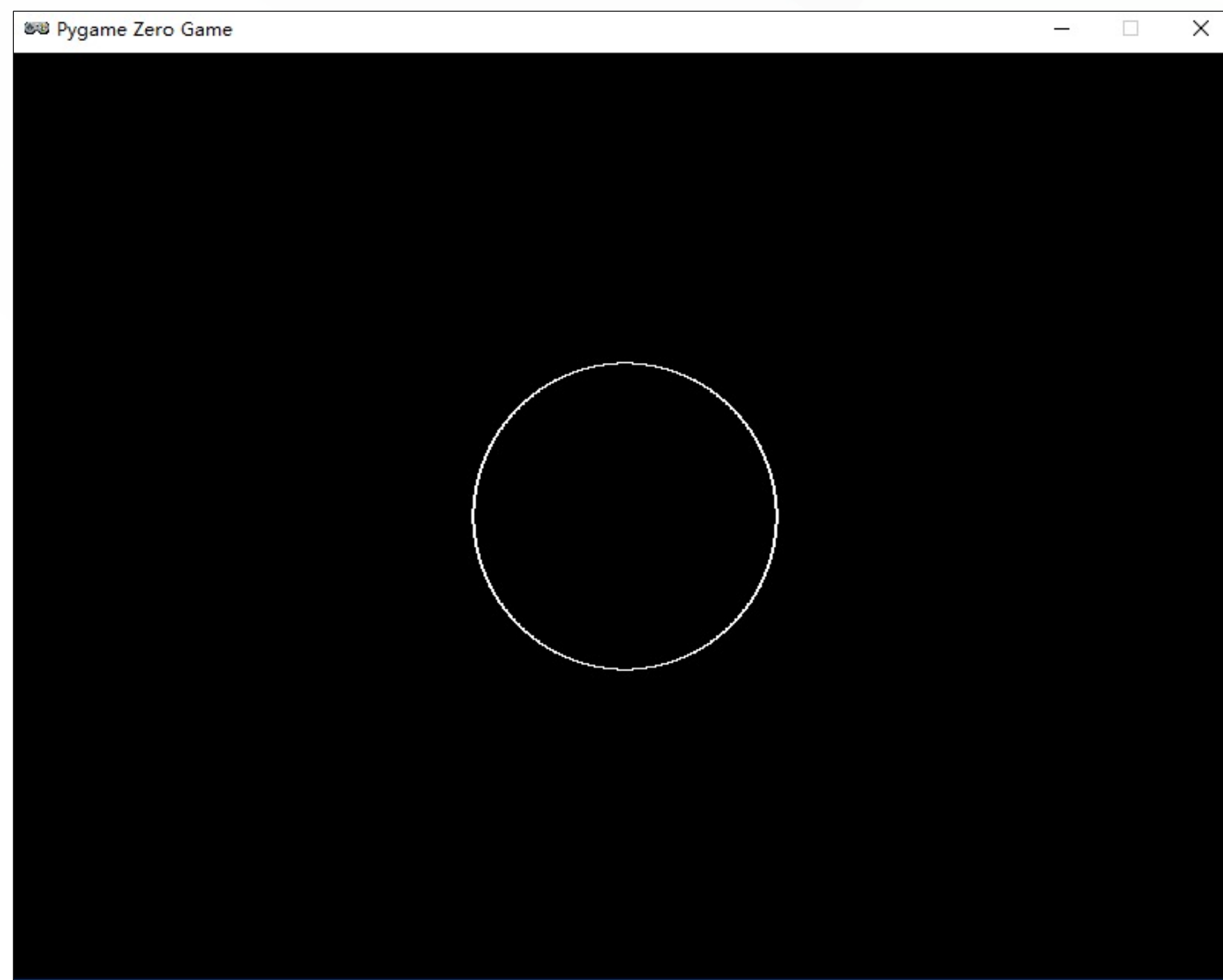
The console at the bottom displays the following output:

```
pygame 1.9.6  
Hello from the pygame community. https://www.pygame.org/contribute.html
```



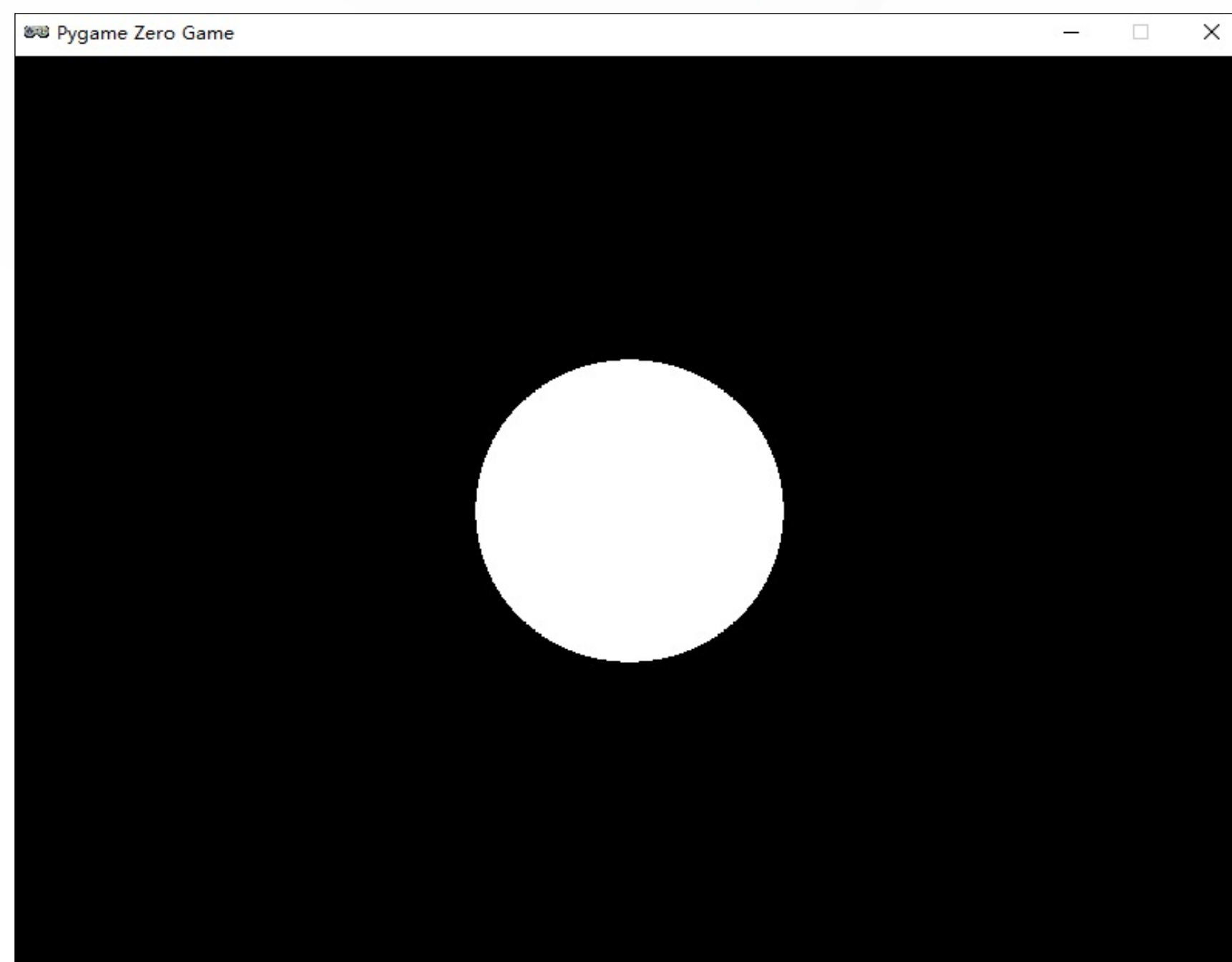
## 项目实践1：弹跳的小球

```
import pgzrun
def draw():
    screen.draw.circle((400, 300), 100, 'white')
pgzrun.go()
```



## 项目实践1：弹跳的小球

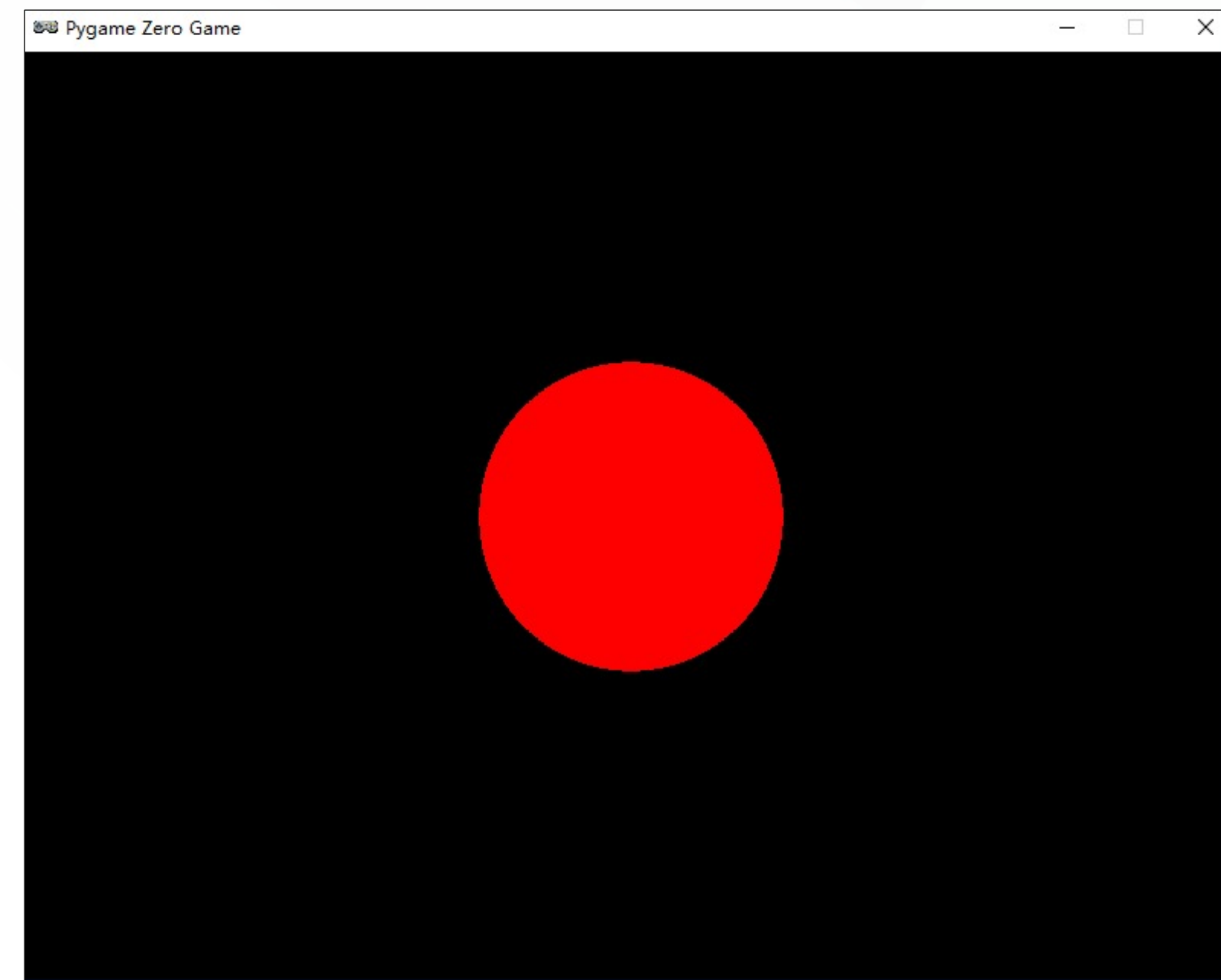
```
import pgzrun
def draw():
    screen.draw.filled_circle((400, 300), 100, 'white')
pgzrun.go()
```





## 项目实践1：弹跳的小球

```
import pgzrun
def draw():
    screen.draw.filled_circle((400, 300), 100, 'red')
pgzrun.go()
```

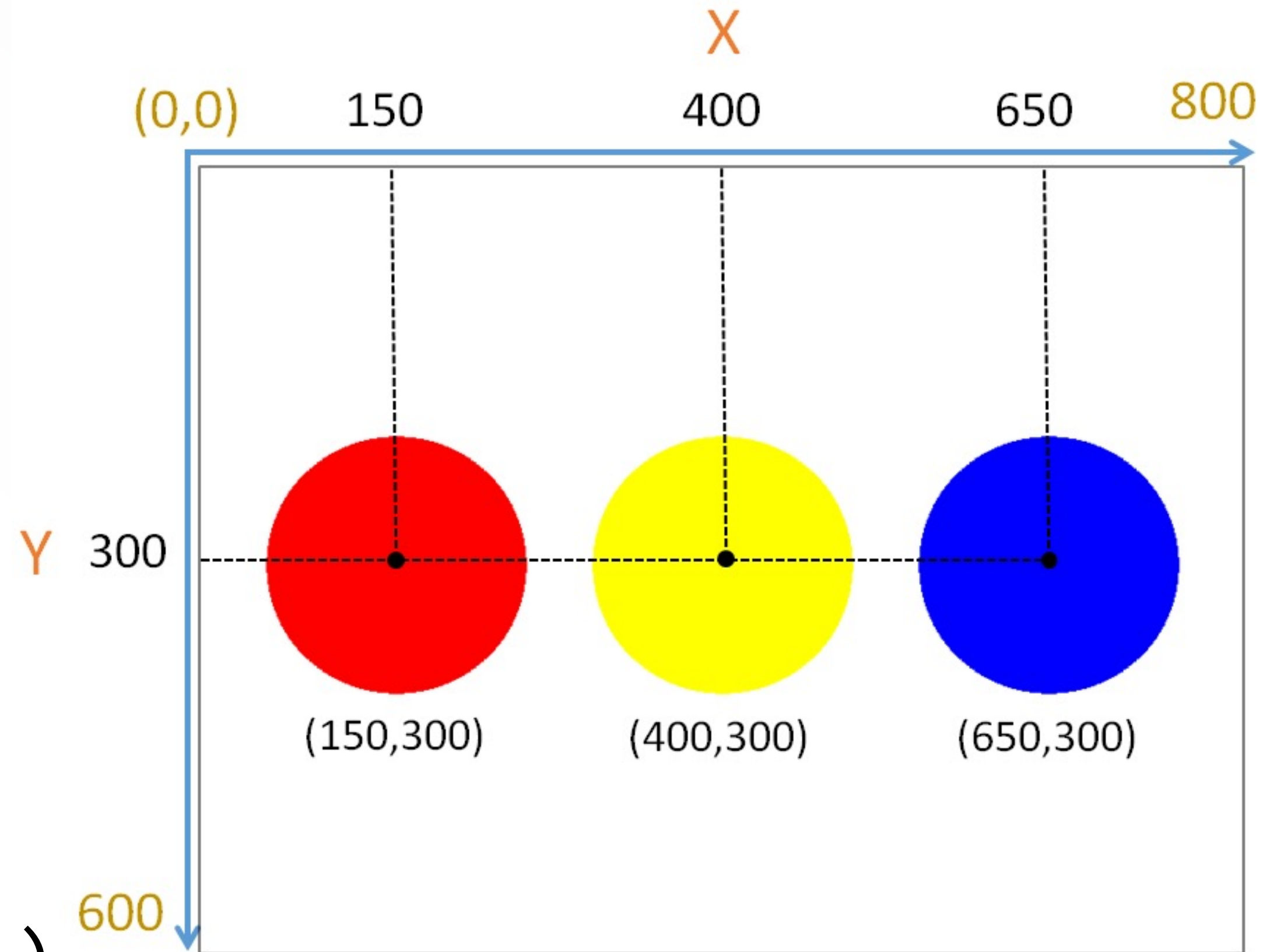


## 项目实践1：弹跳的小球

```
import pgzrun
def draw():
    screen.fill('white')
    screen.draw.filled_circle((400, 300), 100, 'green')
pgzrun.go()
```

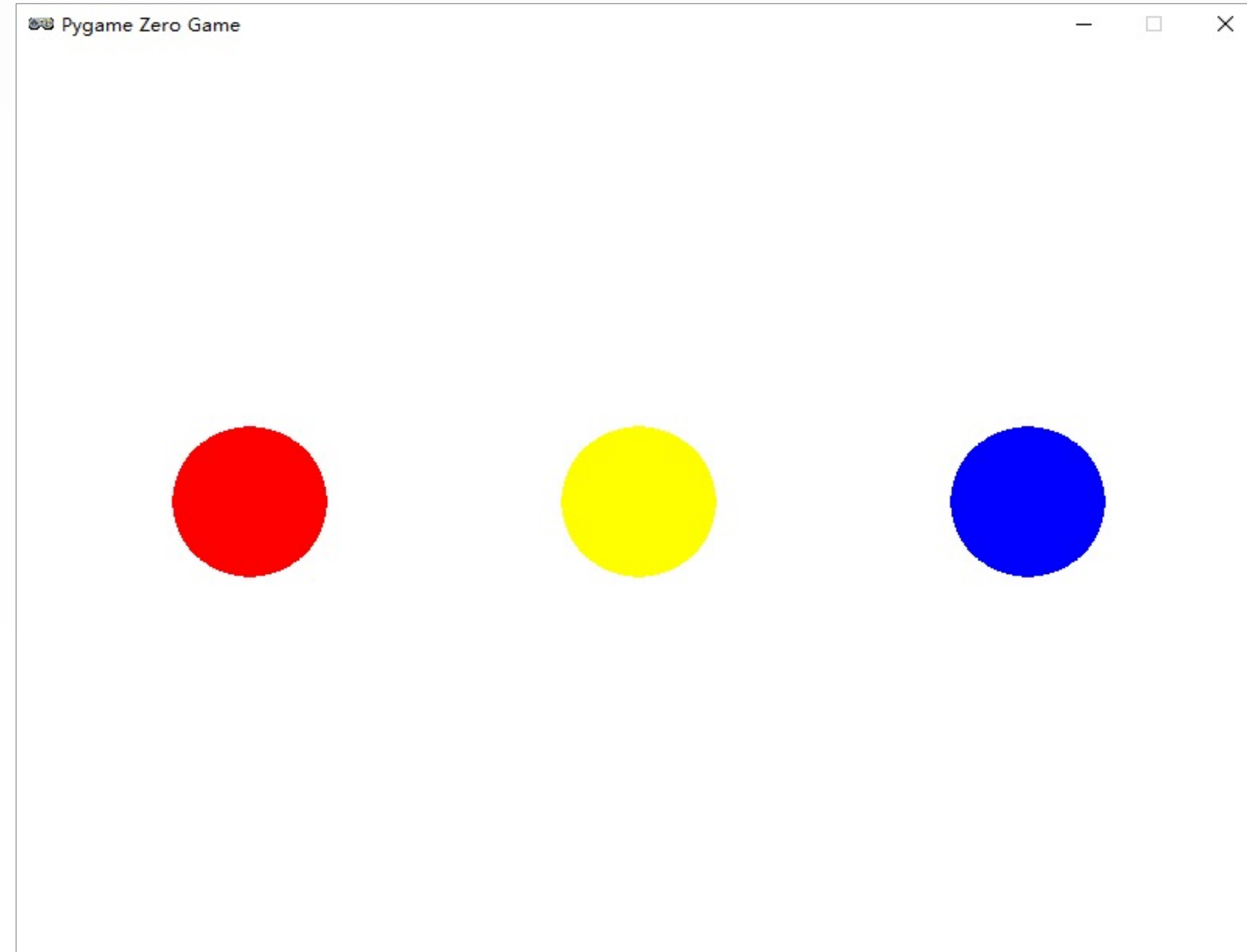


# 项目实践1：弹跳的小球



```
import pgzrun
def draw():
    screen.fill('white')
    screen.draw.filled_circle((150, 300), 100, 'red')
    screen.draw.filled_circle((400, 300), 100, 'yellow')
    screen.draw.filled_circle((650, 300), 100, 'blue')
pgzrun.go()
```

# 项目实践1：弹跳的小球



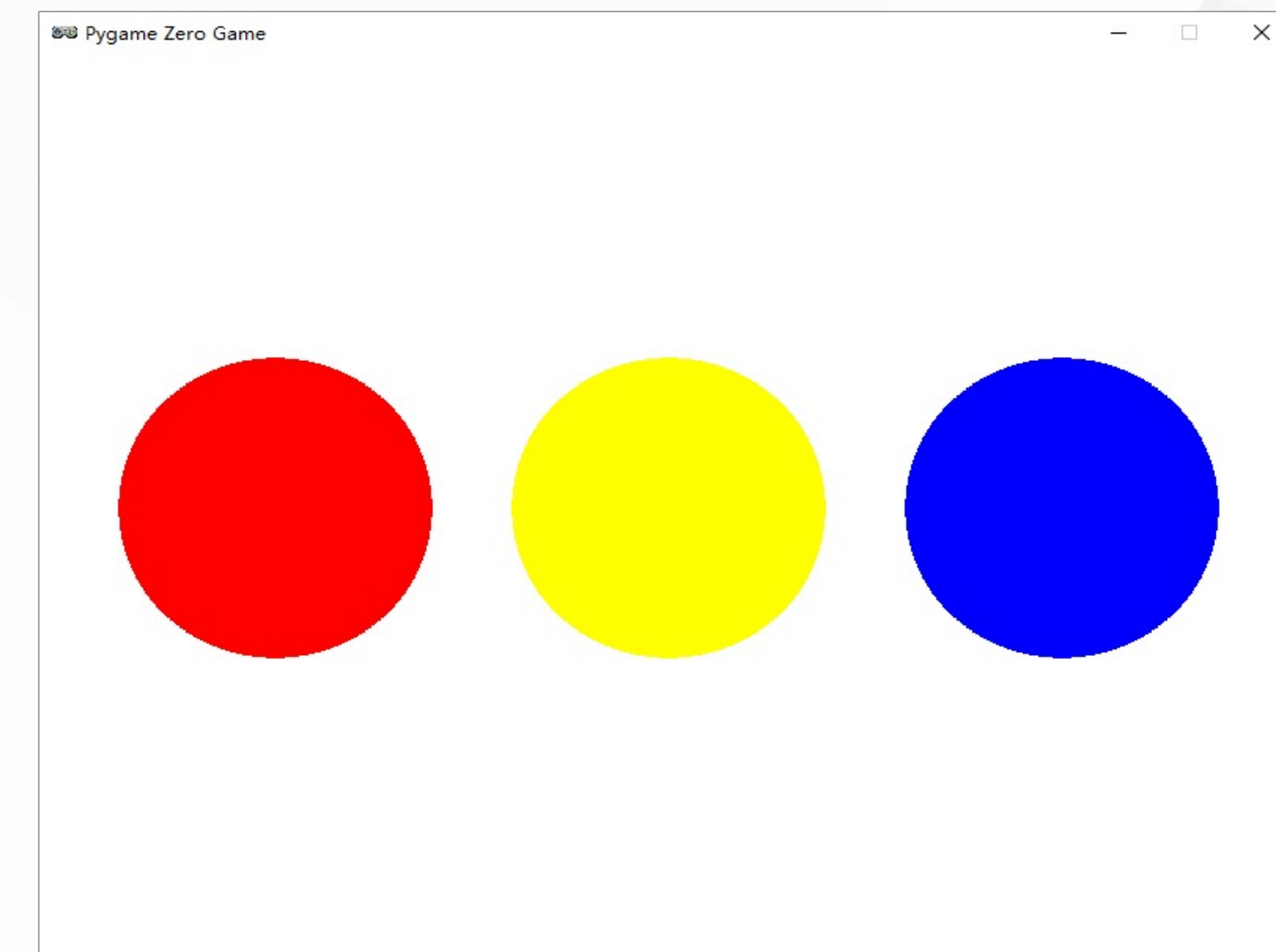
```
import pgzrun
def draw():
    screen.fill('white')
    screen.draw.filled_circle((150, 300), 50, 'red')
    screen.draw.filled_circle((400, 300), 50, 'yellow')
    screen.draw.filled_circle((650, 300), 50, 'blue')
pgzrun.go()
```

# 项目实践1：弹跳的小球

使用变量记录小球半径

2-5-5.py

```
import pgzrun
r = 100
def draw():
    screen.fill('white')
    screen.draw.filled_circle((150, 300), r, 'red')
    screen.draw.filled_circle((400, 300), r, 'yellow')
    screen.draw.filled_circle((650, 300), r, 'blue')
pgzrun.go()
```



- $r=125$  即可同时修改三个球的半径大小

## 项目实践1：弹跳的小球

2-5-1.py

---

```
r = 10  
print(r)
```

---

2-5-2.py

---

```
str = 'Python 真好玩'  
print(str)
```

---

2-5-3.py

---

```
r = 1  
print(r)  
r = 2  
print(r)  
t = r  
print(t)
```

---

2-5-4.py

---

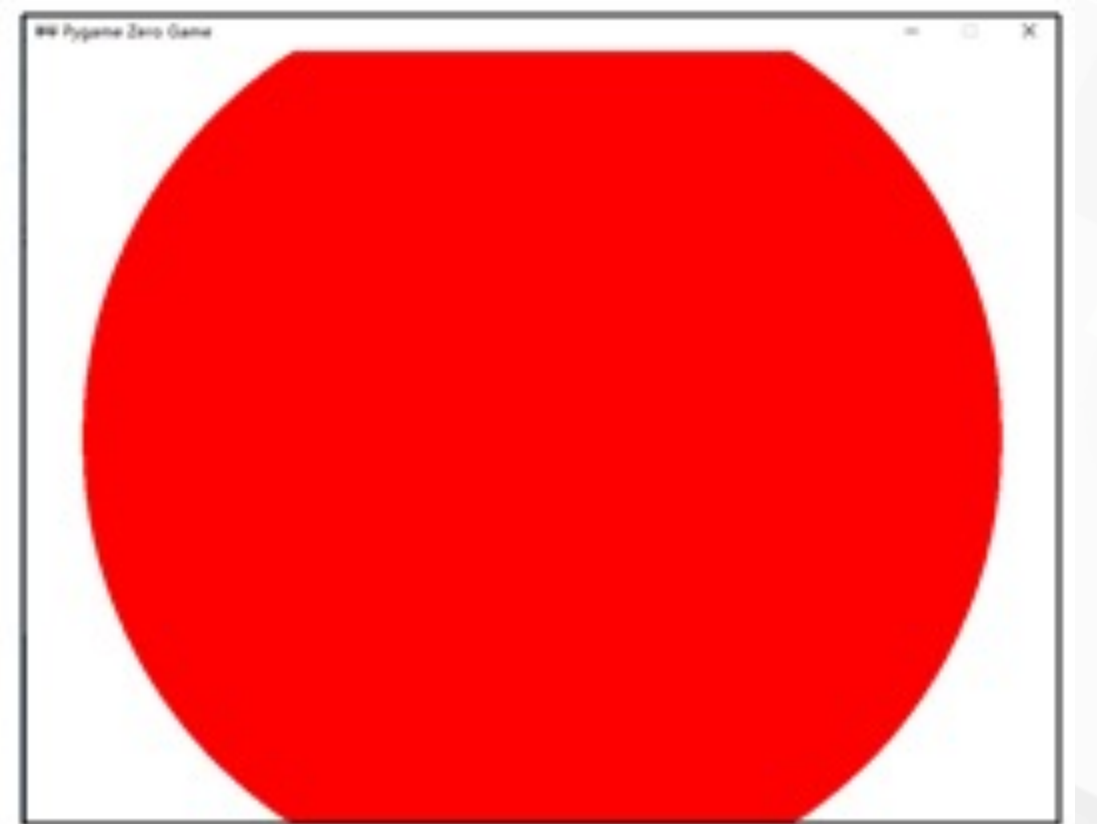
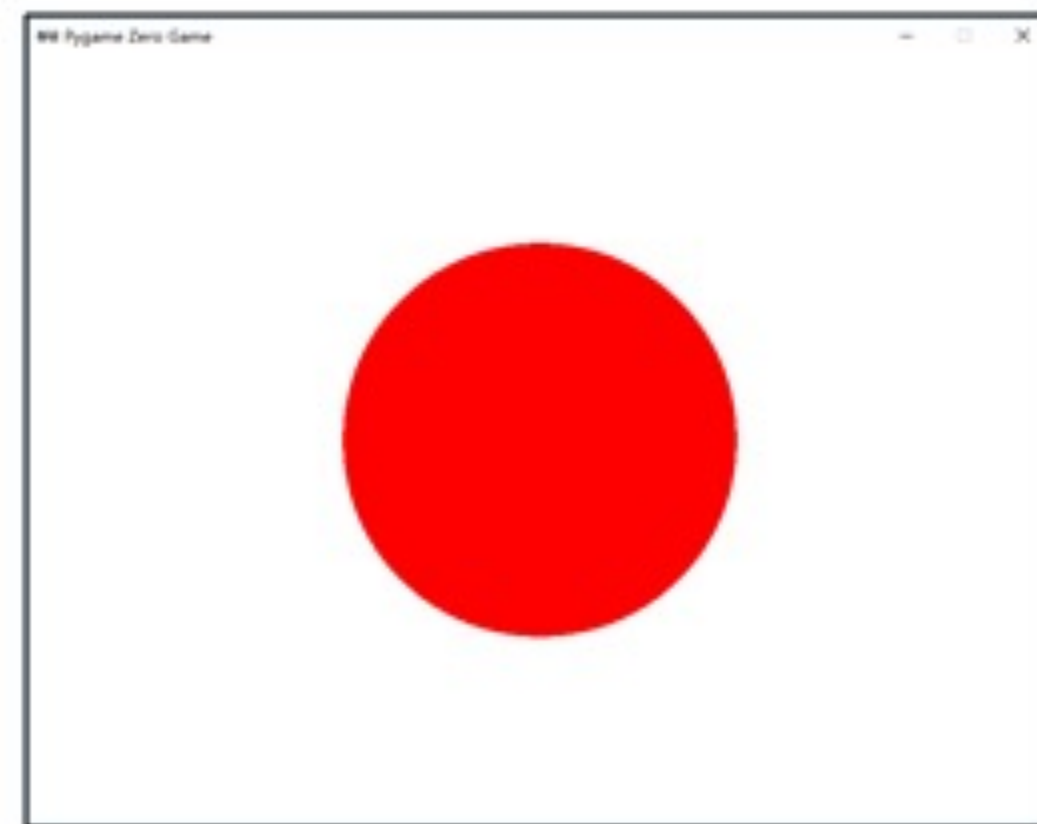
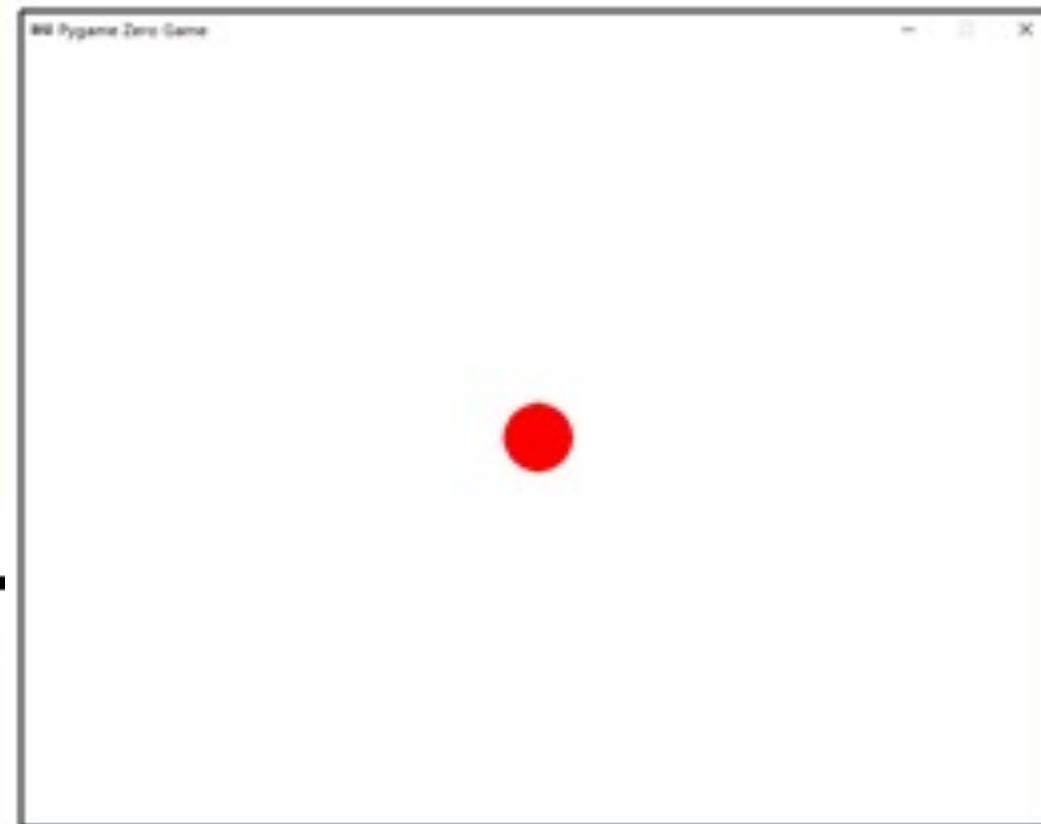
```
r = 1  
print(r)  
r = r+2  
print(r)  
t = r-1  
print(t)  
t = t*3  
print(t)  
t = t/2  
print(t)
```

---

## 项目实践1：弹跳的小球

2-6.py

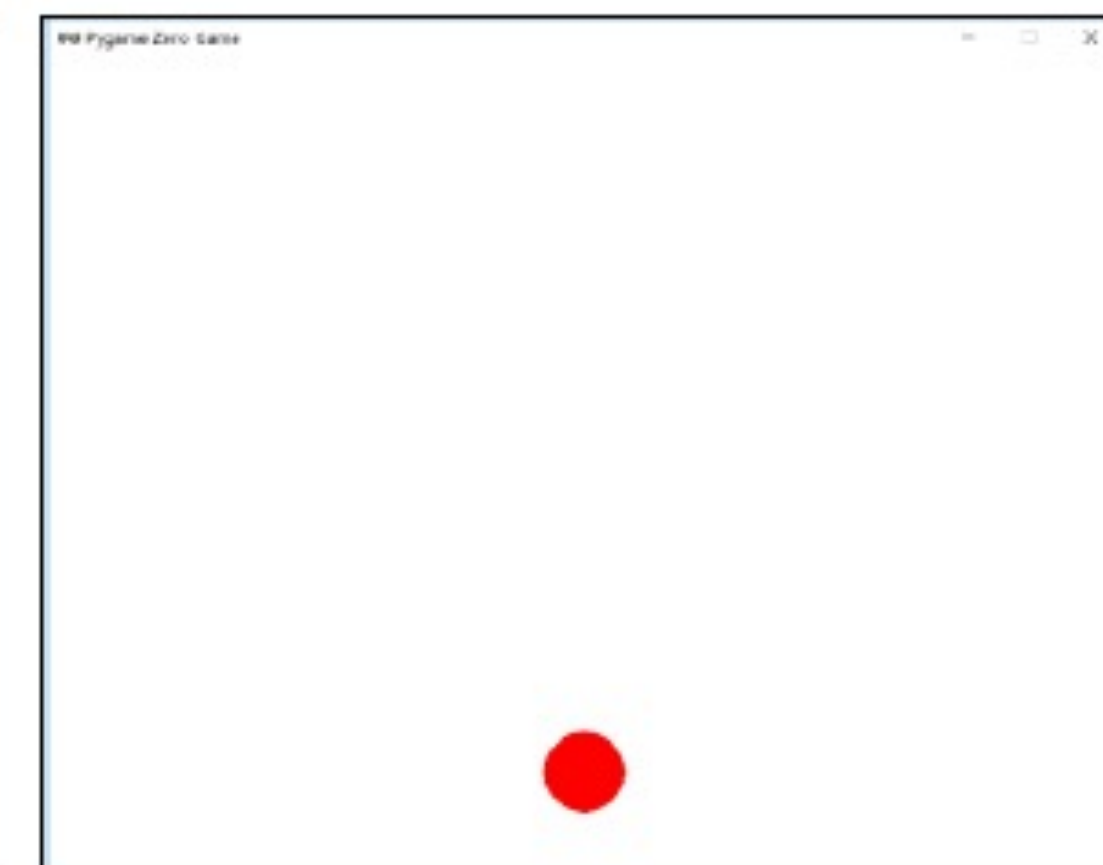
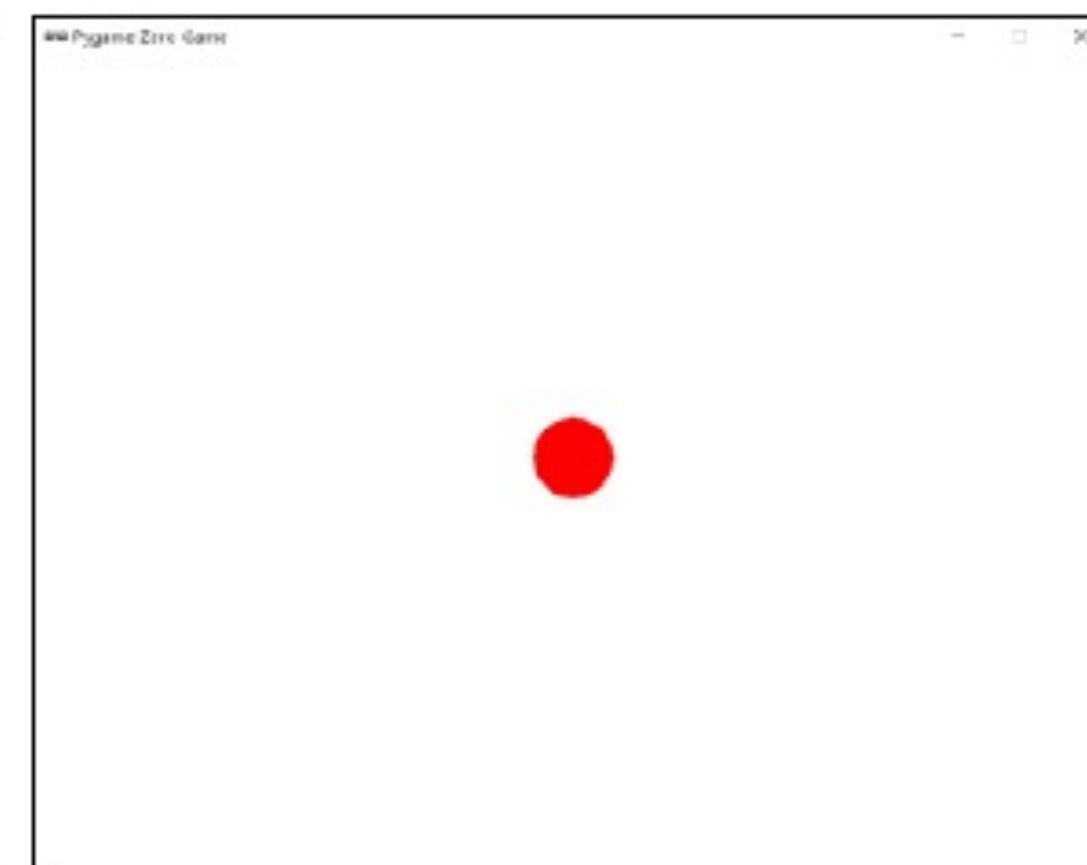
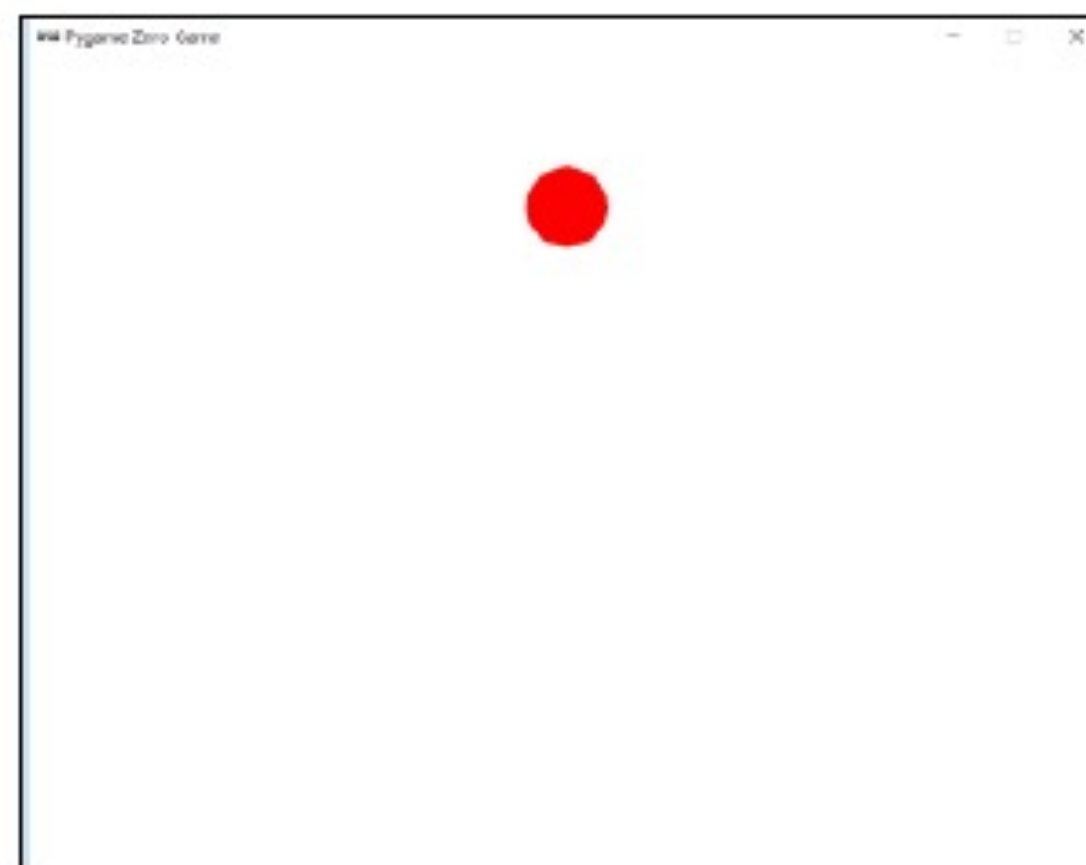
```
import pgzrun
r = 1
def draw():
    screen.fill('white')
    screen.draw.filled_circle((400, 300), r, 'red')
def update():
    global r
    r = r+1
pgzrun.go()
```



# 项目实践1：弹跳的小球

2-7.py

```
import pgzrun
y = 100
def draw():
    screen.fill('white')
    screen.draw.filled_circle((400, y), 30, 'red')
def update():
    global y
    y = y+1
pgzrun.go()
```



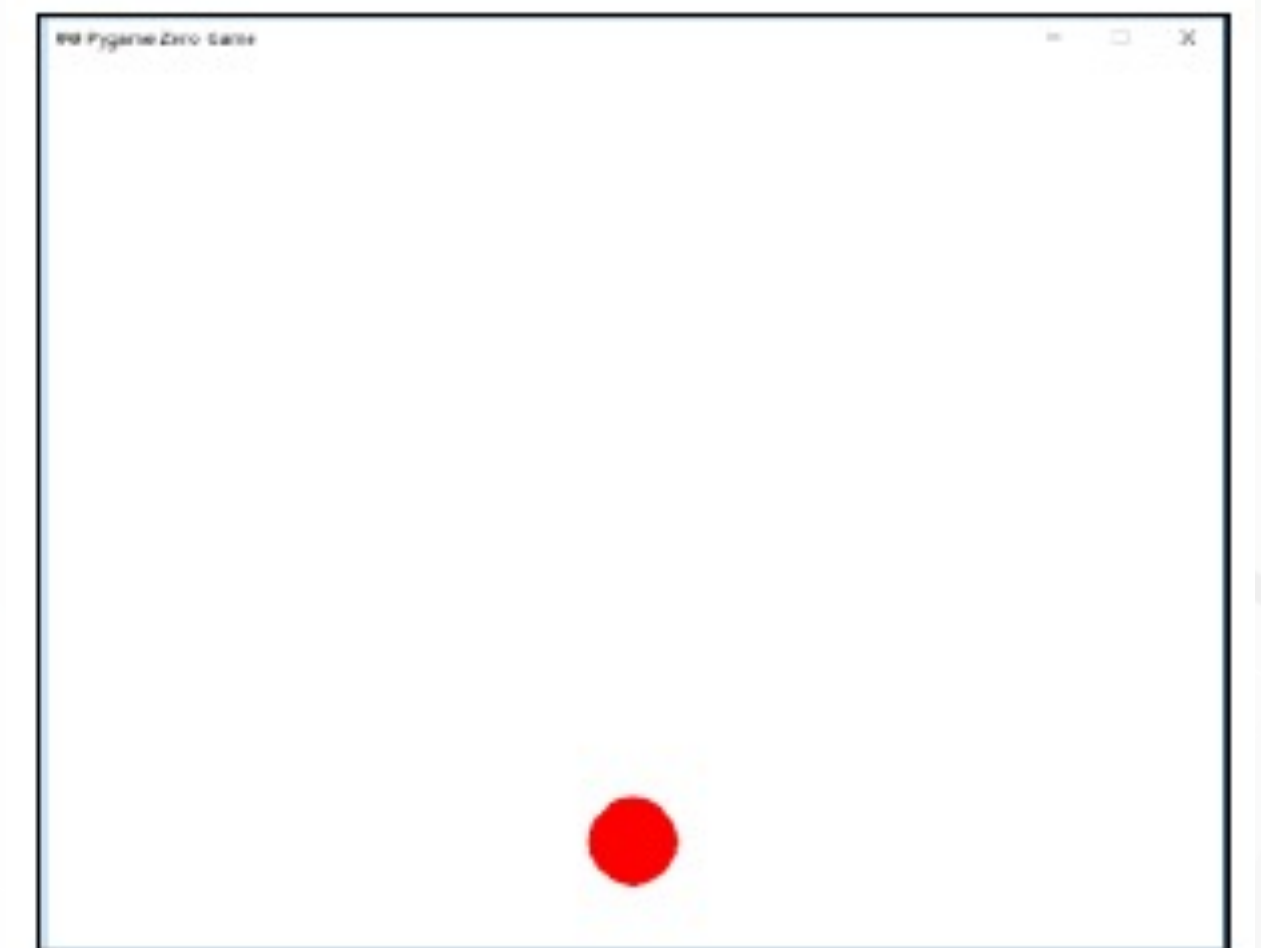


# 项目实践1：弹跳的小球

利用if语句实现小球重复下落

2-8-1.py

```
import pgzrun
y = 100
def draw():
    screen.fill('white')
    screen.draw.filled_circle((400, y), 30, 'red')
def update():
    global y
    y = y+3
    if y>630:
        y = 0
pgzrun.go()
```



## 项目实践1：弹跳的小球

2-8-2.py

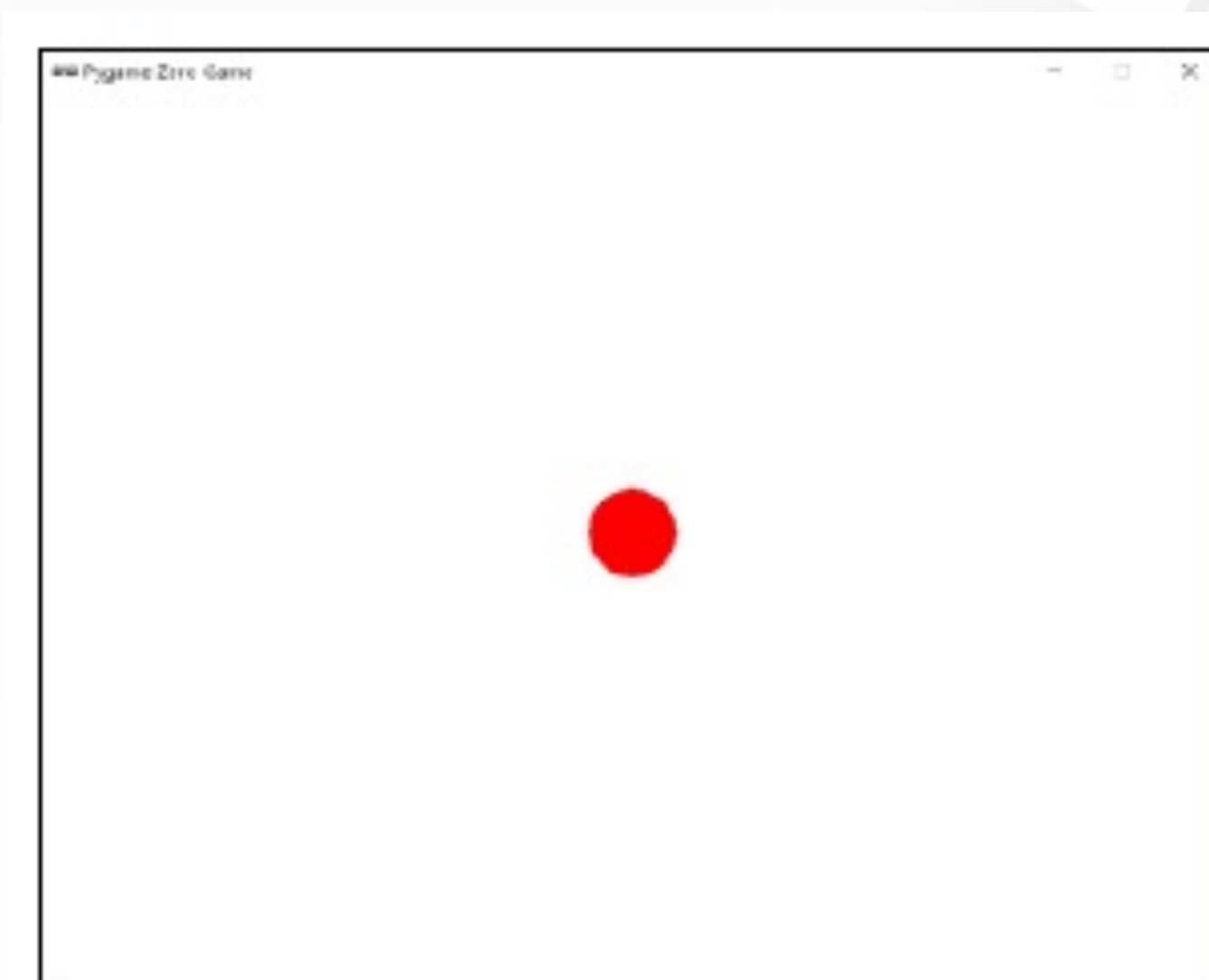
```
x = 3
y = 5
if x > y:
    print('x 大')
if x == y:
    print('x 与 y 一样大')
if x < y:
    print('y 大')
```

表达式	含义
$x > y$	x 是否大于 y
$x < y$	x 是否小于 y
$x == y$	x 是否等于 y
$x != y$	x 是否不等于 y
$x \geq y$	x 是否大于或等于 y
$x \leq y$	x 是否小于或等于 y

## 项目实践1：弹跳的小球

2-9-1.py

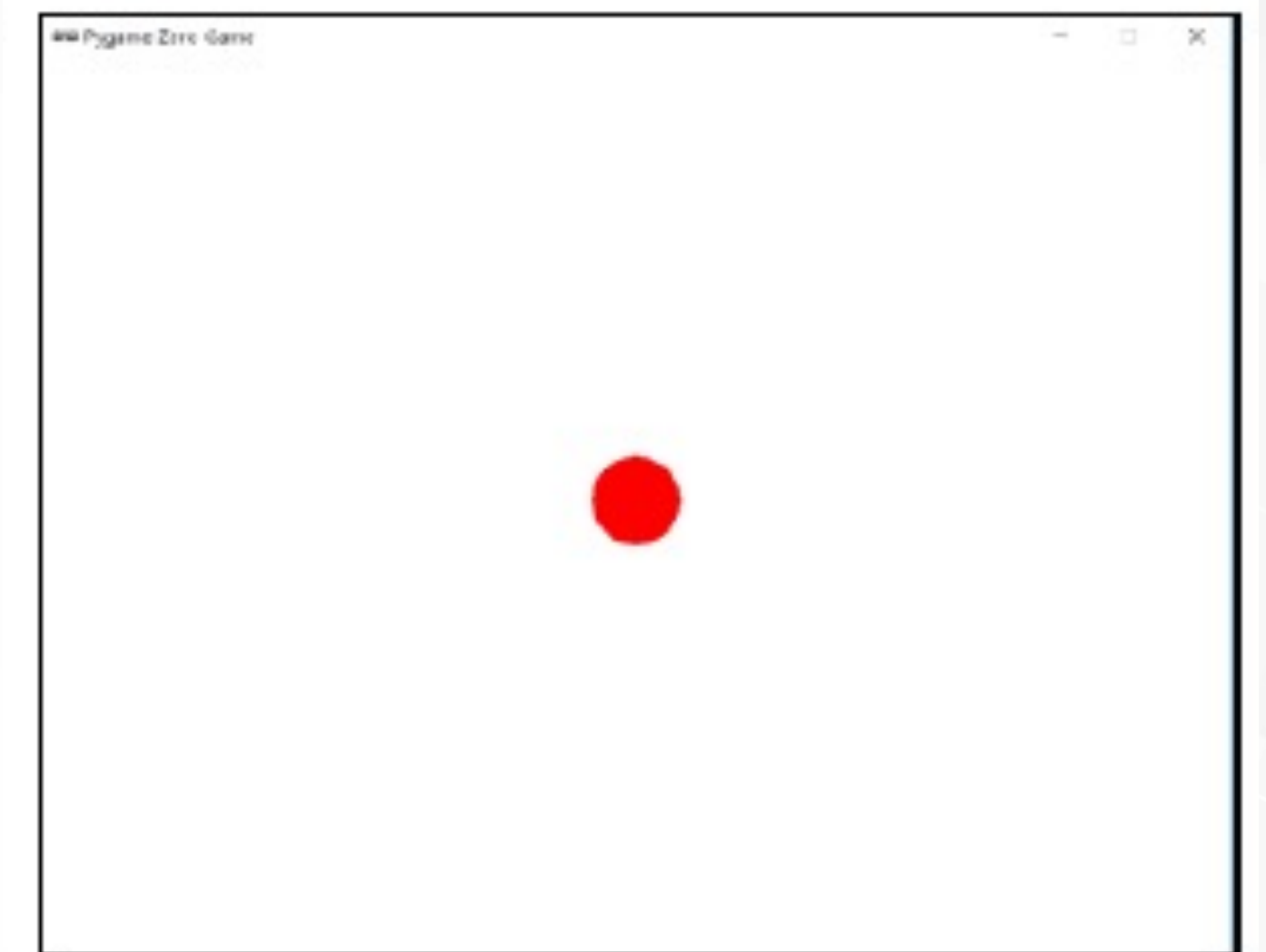
```
import pgzrun
y = 100
speed_y = 3
def draw():
    screen.fill('white')
    screen.draw.filled_circle((400, y), 30, 'red')
def update():
    global y
    y = y+speed_y
pgzrun.go()
```



# 项目实践1：弹跳的小球

2-9-2.py

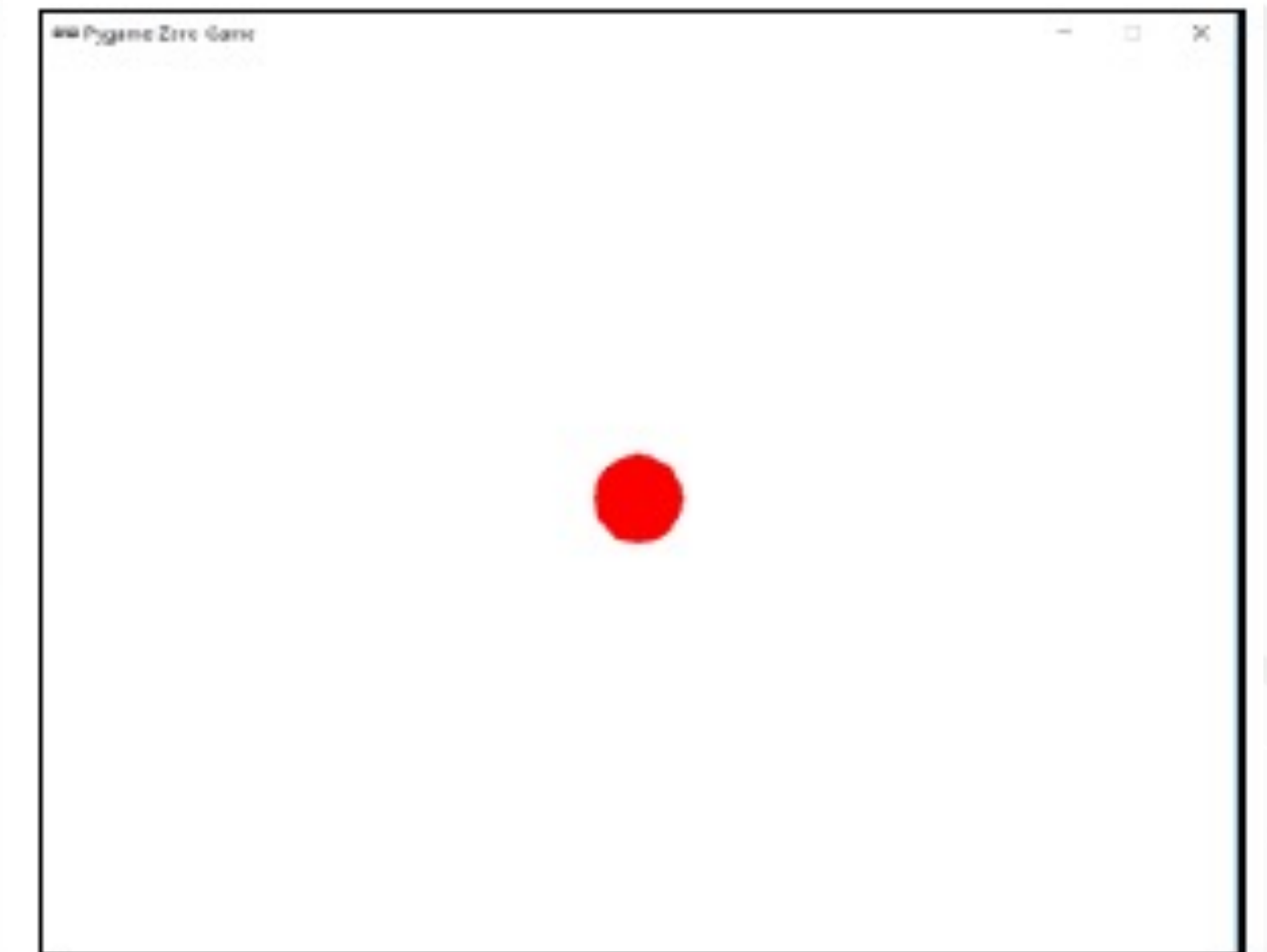
```
import pgzrun
y = 100
speed_y = 3
def draw():
    screen.fill('white') # 白色背景
    screen.draw.filled_circle((400, y), 30, 'red')
def update():
    global y, speed_y
    y = y+speed_y
    if y>=570:
        speed_y = -speed_y
pgzrun.go()
```



# 项目实践1：弹跳的小球

2-9-3.py

```
import pgzrun
y = 100
speed_y = 3
def draw():
    screen.fill('white')
    screen.draw.filled_circle((400, y), 30, 'red')
def update():
    global y, speed_y
    y = y+speed_y
    if y >= 570:
        speed_y = -speed_y
    if y <= 30:
        speed_y = -speed_y
pgzrun.go()
```



# 项目实践1：弹跳的小球

2-9-4.py

```
print(3 > 2)  
print(4 > 5)
```

2-9-5.py

```
print(not(3 > 2))  
print(not(4 > 5))
```

2-9-6.py

```
print((3 > 2) or (4 > 5))  
print((3 > 2) and (4 > 5))
```

## 项目实践1：弹跳的小球

2-9-7.py

```
if y >= 570:  
    speed_y = -speed_y  
if y <= 30:  
    speed_y = -speed_y
```

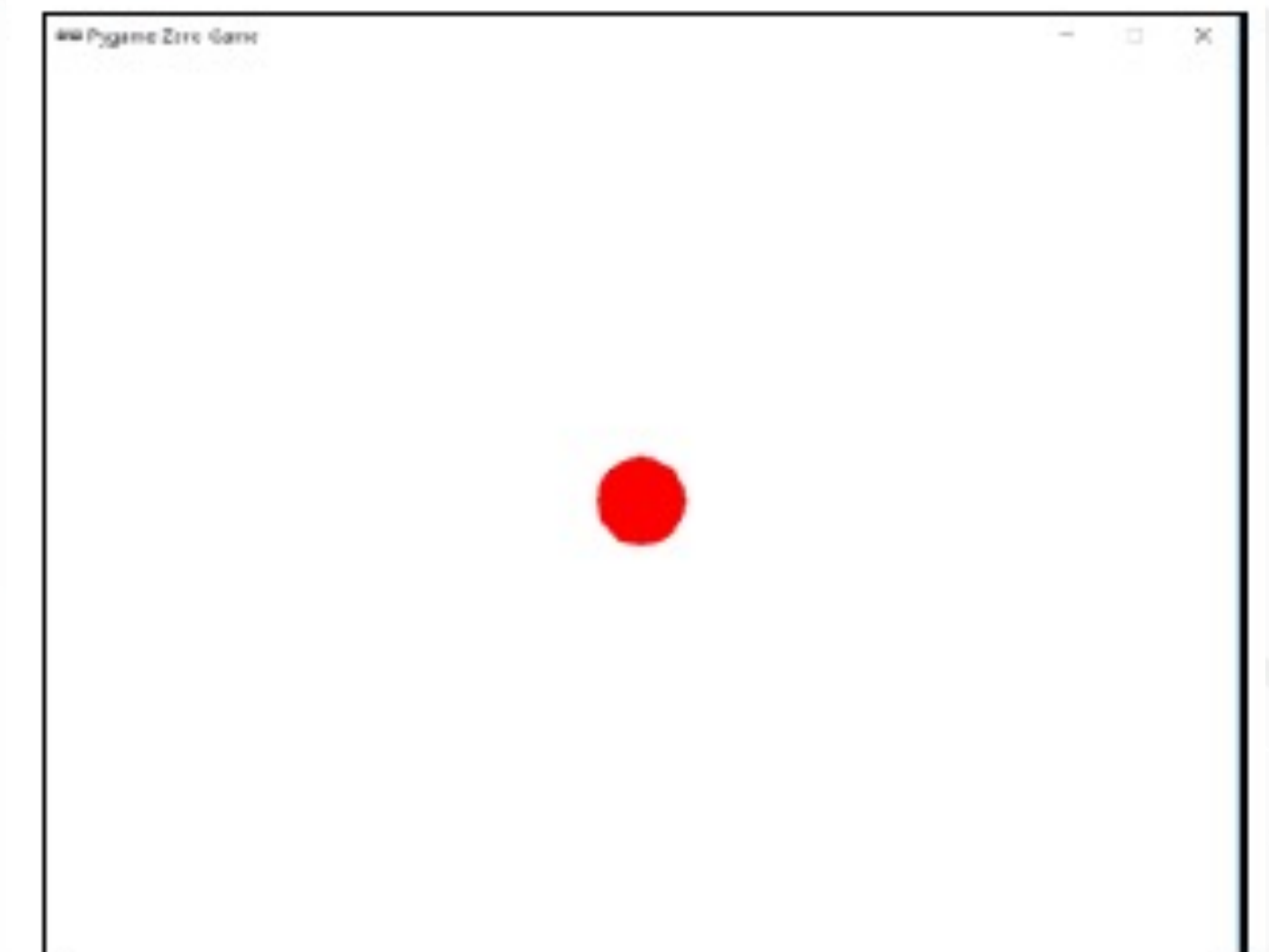


```
import pgzrun  
y = 100  
speed_y = 3  
def draw():  
    screen.fill('white')  
    screen.draw.filled_circle((400, y), 30, 'red')  
def update():  
    global y, speed_y  
    y = y+speed_y  
    if y>=570 or y<=30:  
        speed_y = -speed_y  
pgzrun.go()
```

# 项目实践1：弹跳的小球

2-9-8.py

```
import pgzrun
HEIGHT = 600
WIDTH = 800
y = 100
speed_y = 3
r = 30
def draw():
    screen.fill('white')
    screen.draw.filled_circle((WIDTH/2, y), r, 'red')
def update():
    global y, speed_y
    y = y+speed_y
    if y >= HEIGHT-r or y <= r:
        speed_y = -speed_y
pgzrun.go()
```



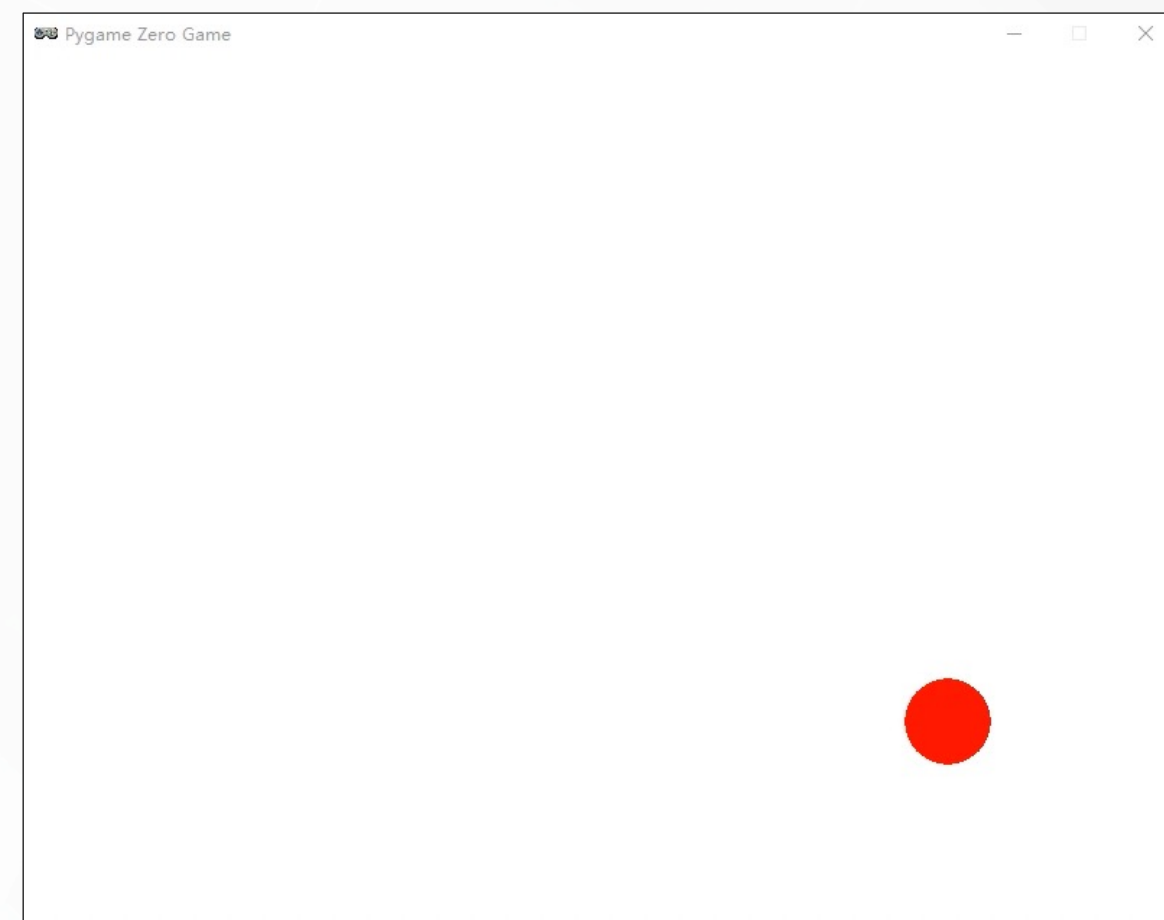


## 项目实践1：弹跳的小球

2-10-2.py

```
import pgzrun
WIDTH = 800
HEIGHT = 600
x = WIDTH/2
y = HEIGHT/2
speed_x = 3
speed_y = 5
r = 30
def draw():
    screen.fill('white')
    screen.draw.filled_circle((x, y), r, 'red')
def update():
```

```
    def update():
        global x,y,speed_x,speed_y
        x = x+speed_x
        y = y+speed_y
        if x >= WIDTH-r or x <= r:
            speed_x = -speed_x
        if y >= HEIGHT-r or y <= r:
            speed_y = -speed_y
pgzrun.go()
```



# 项目实践1：弹跳的小球

2-10-3.py

```
import pgzrun # 导入游戏库

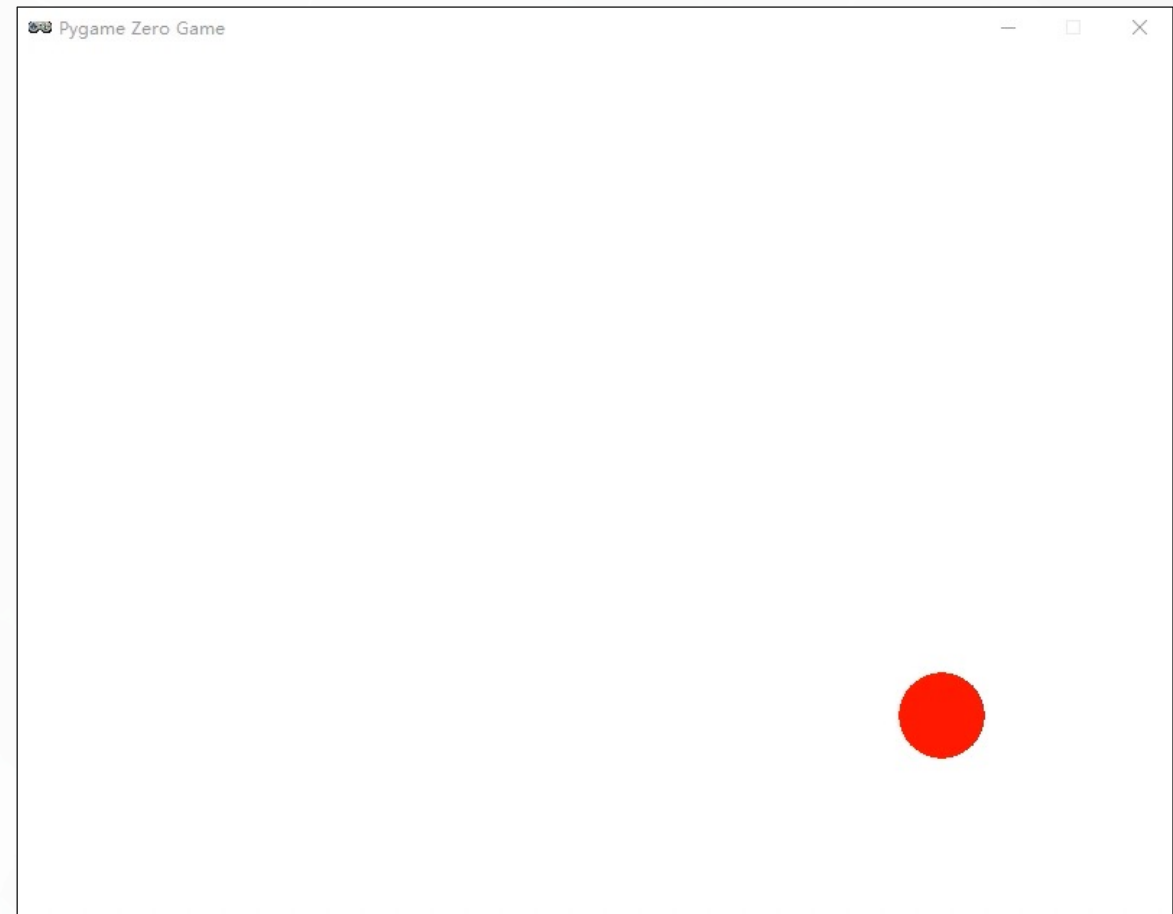
WIDTH = 800 # 设置窗口的宽度
HEIGHT = 600 # 设置窗口的高度
x = WIDTH/2 # 小球的 x 坐标, 初始化在窗口中间
y = HEIGHT/2 # 小球的 y 坐标, 初始化在窗口中间
speed_x = 3 # 小球 x 方向的速度
speed_y = 5 # 小球 y 方向的速度
r = 30 # 小球的半径

def draw(): # 绘制模块, 每帧重复执行
    screen.fill('white') # 白色背景
    # 绘制一个填充圆, 坐标(x,y), 半径 r, 红色
    screen.draw.filled_circle((x, y), r, 'red')
```

## 自由落体的小球

```
def update(): # 更新模块, 每帧重复操作
    global x,y,speed_x,speed_y # 要修改的变量在这里说明下
    x = x+speed_x # 利用 x 方向速度更新 x 坐标
    y = y+speed_y # 利用 y 方向速度更新 y 坐标
    if x >= WIDTH-r or x <= r: # 当小球碰到左右边界时
        speed_x = -speed_x # x 方向速度反转
    if y >= HEIGHT-r or y <= r: # 当小球碰到上下边界时
        speed_y = -speed_y # y 方向速度反转

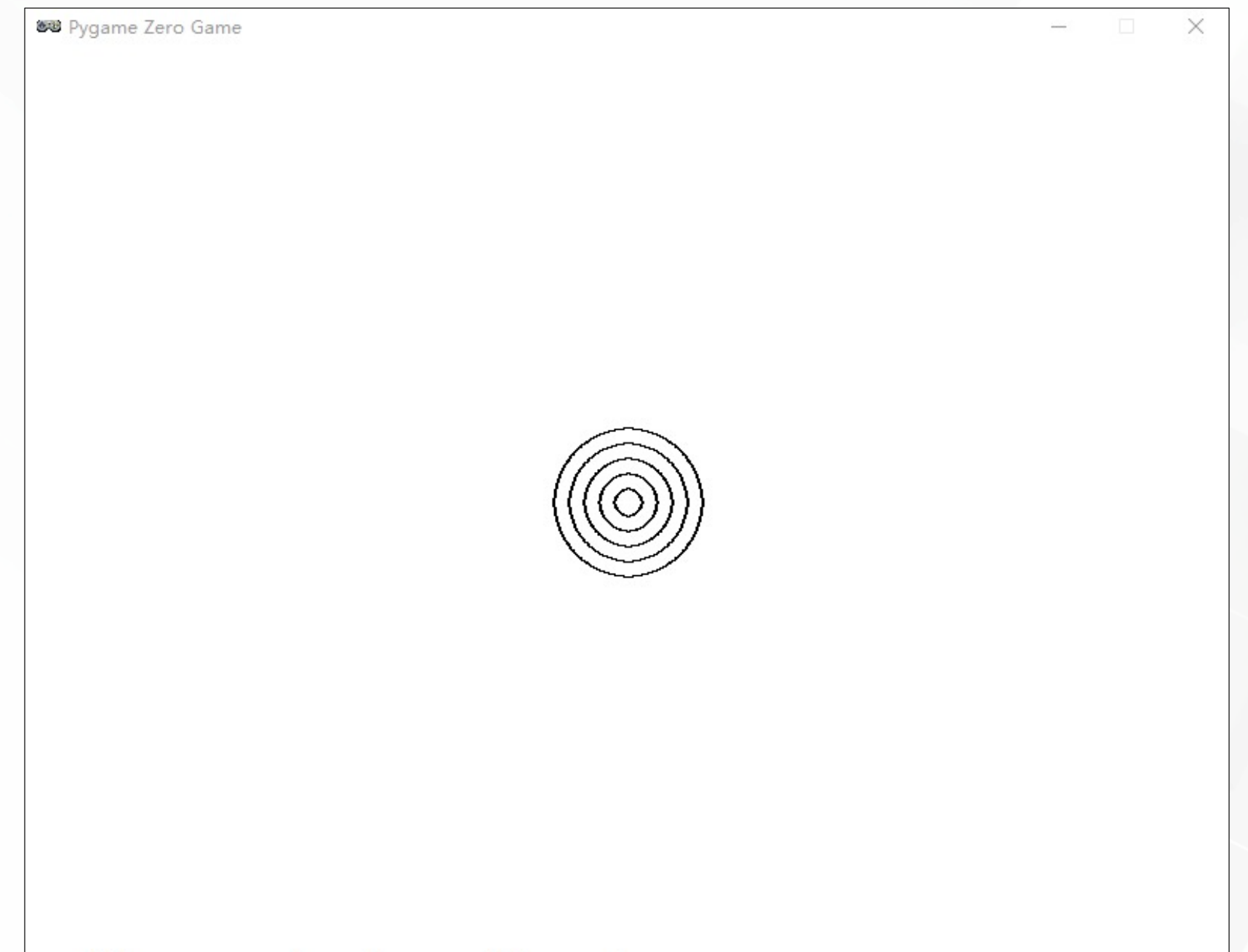
pgzrun.go() # 开始执行游戏
```



## 项目实践2：绘制同心圆

3-1-1.py

```
import pgzrun
def draw():
    screen.fill('white')
    screen.draw.circle((400, 300), 10, 'black')
    screen.draw.circle((400, 300), 20, 'black')
    screen.draw.circle((400, 300), 30, 'black')
    screen.draw.circle((400, 300), 40, 'black')
    screen.draw.circle((400, 300), 50, 'black')
pgzrun.go()
```

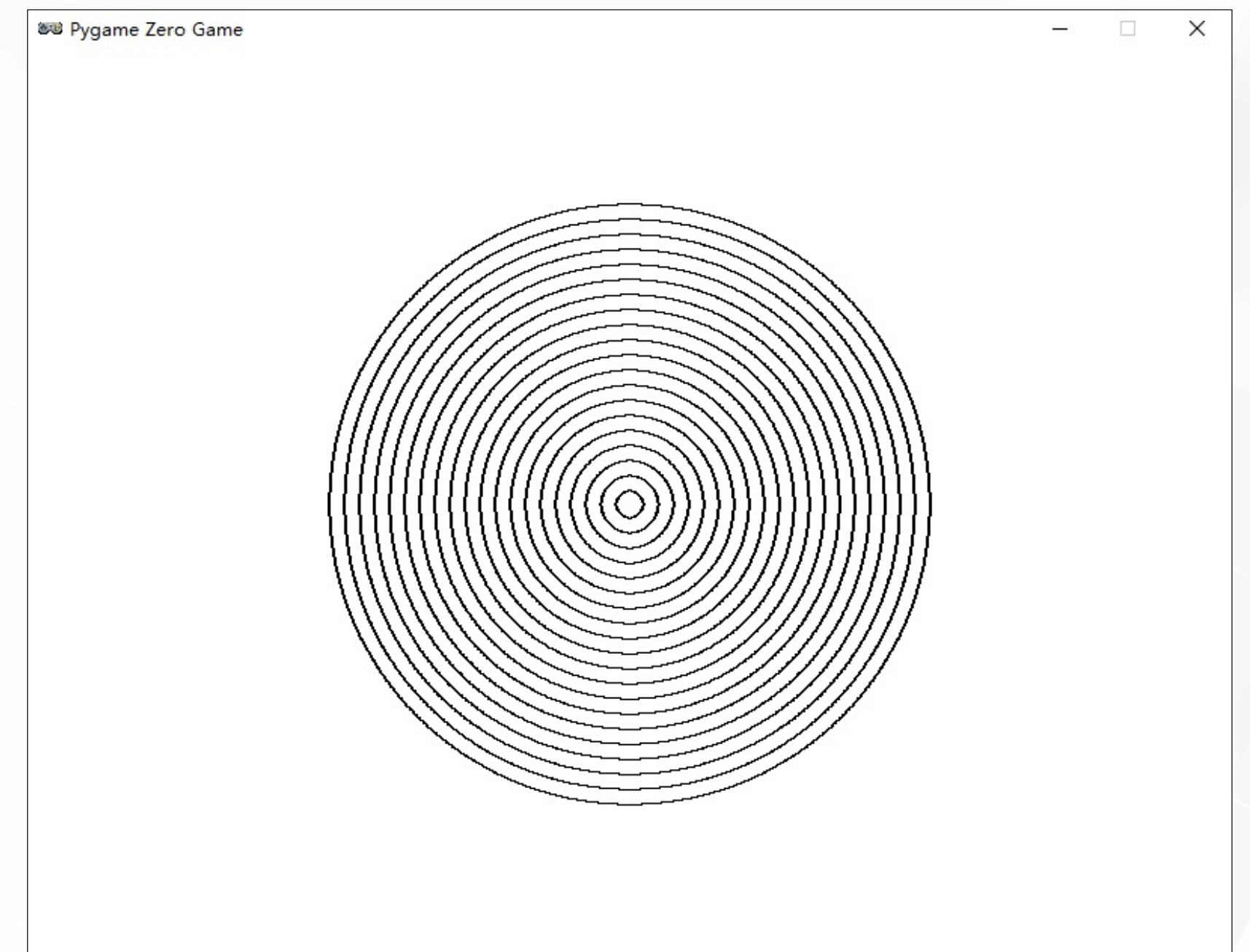


## 项目实践2：绘制同心圆

## 练习：绘制20层同心圆

3-1-1.py

```
import pgzrun
def draw():
    screen.fill('white')
    screen.draw.circle((400, 300), 10, 'black')
    screen.draw.circle((400, 300), 20, 'black')
    screen.draw.circle((400, 300), 30, 'black')
    screen.draw.circle((400, 300), 40, 'black')
    screen.draw.circle((400, 300), 50, 'black')
pgzrun.go()
```

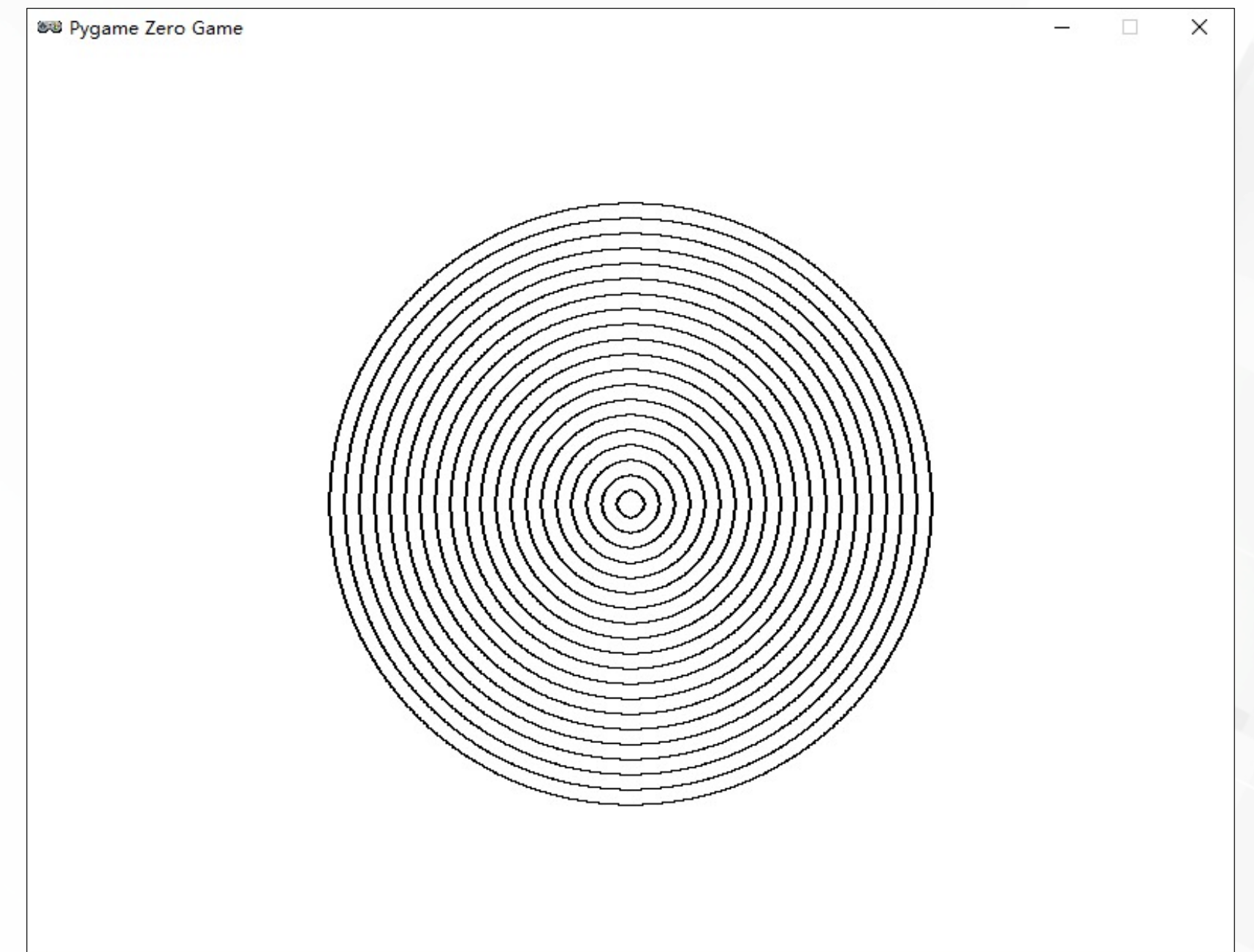


## 项目实践2：绘制同心圆

用for循环语句绘制20层同心圆

3-1-2.py

```
import pgzrun
def draw():
    screen.fill('white')
    for r in range(1, 201, 10):
        screen.draw.circle((400, 300), r, 'black')
pgzrun.go()
```



## 项目实践2：绘制同心圆

3-2-1.py

```
for i in range(5):  
    print(i)
```

```
0  
1  
2  
3  
4
```

3-2-2.py

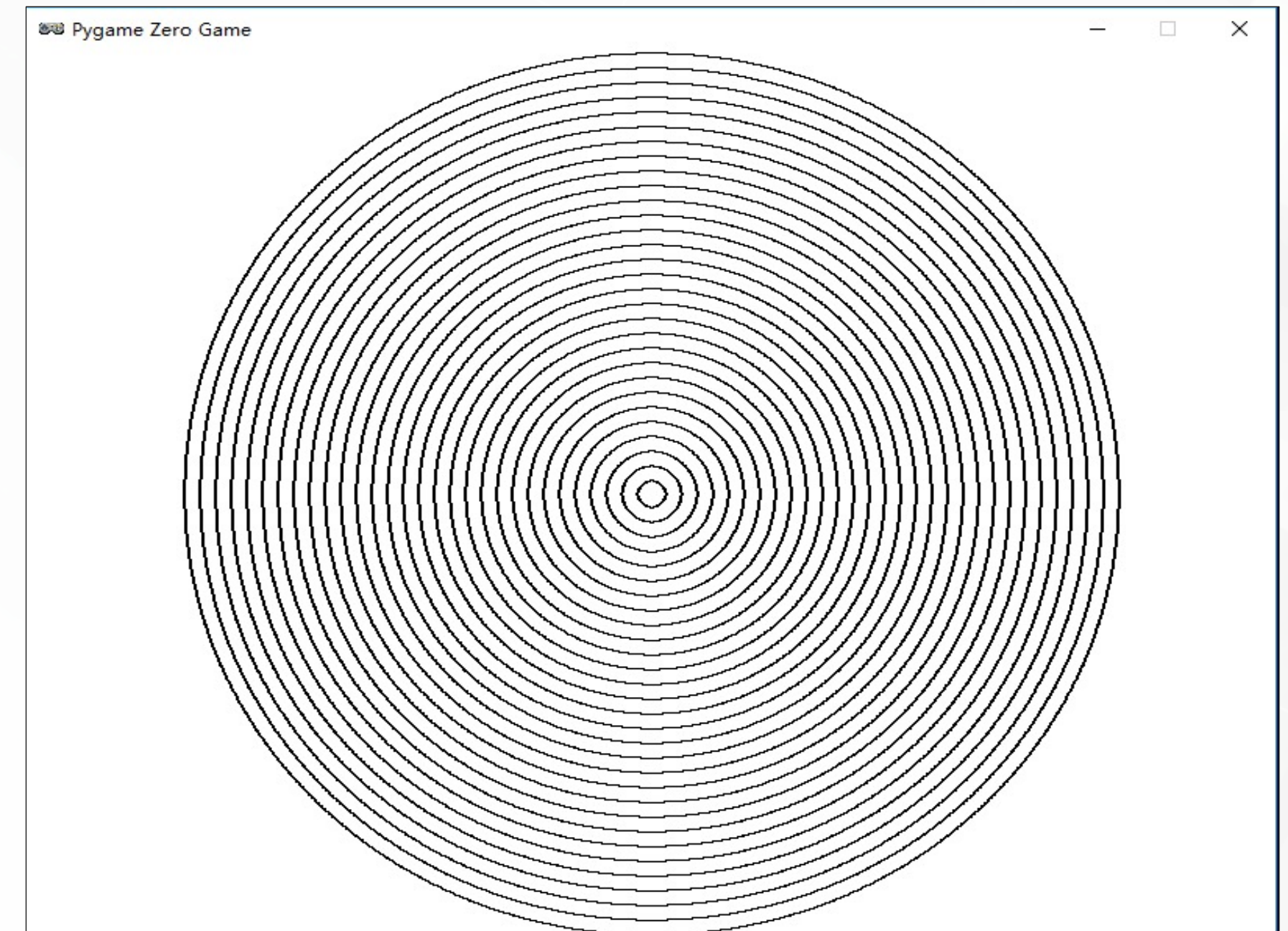
```
for i in range(3,6):  
    print(i)
```

```
3  
4  
5
```

## 项目实践2：绘制同心圆

### 用for语句绘制30层同心圆

```
import pgzrun
def draw():
    screen.fill('white')
    for r in range(1, 31):
        screen.draw.circle((400, 300), 10*r, 'black')
pgzrun.go()
```



## 项目实践2：绘制同心圆

### 3-2-4.py

```
for i in range(1, 11, 2):  
    print(i)
```

```
1  
3  
5  
7  
9
```

3-2-3.py 画 5 层同心圆的代码，也可以修改为：

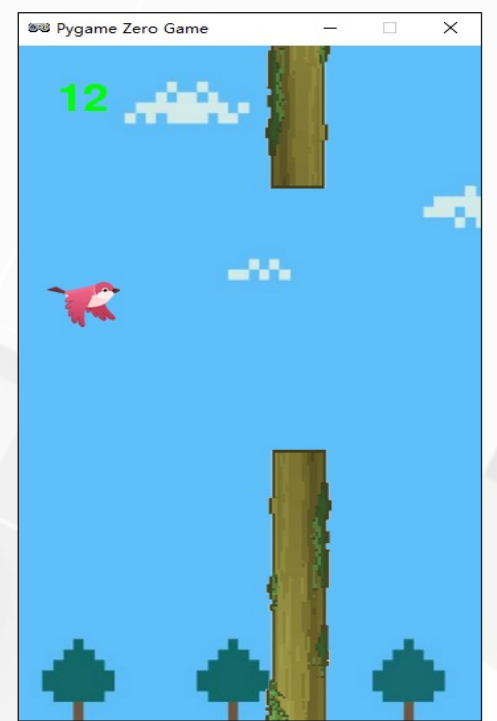
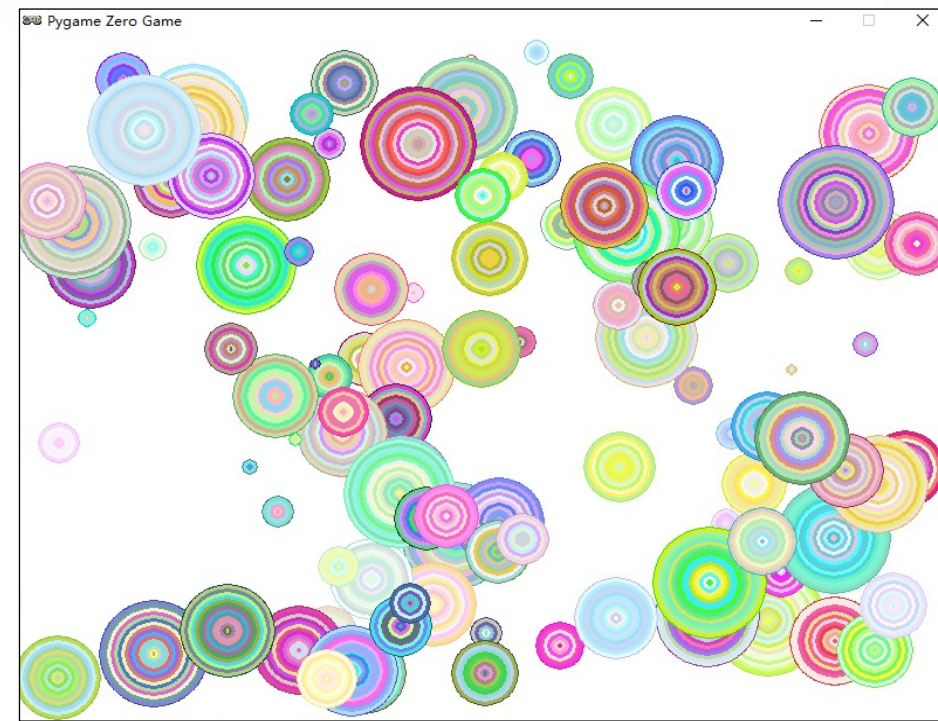
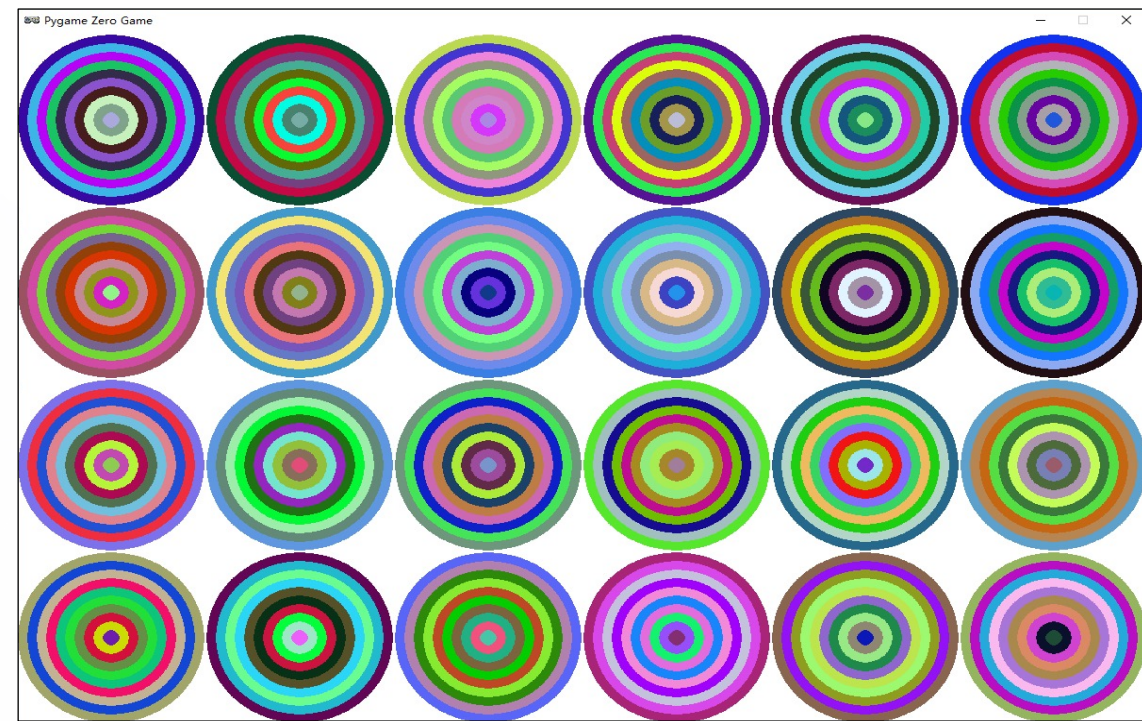
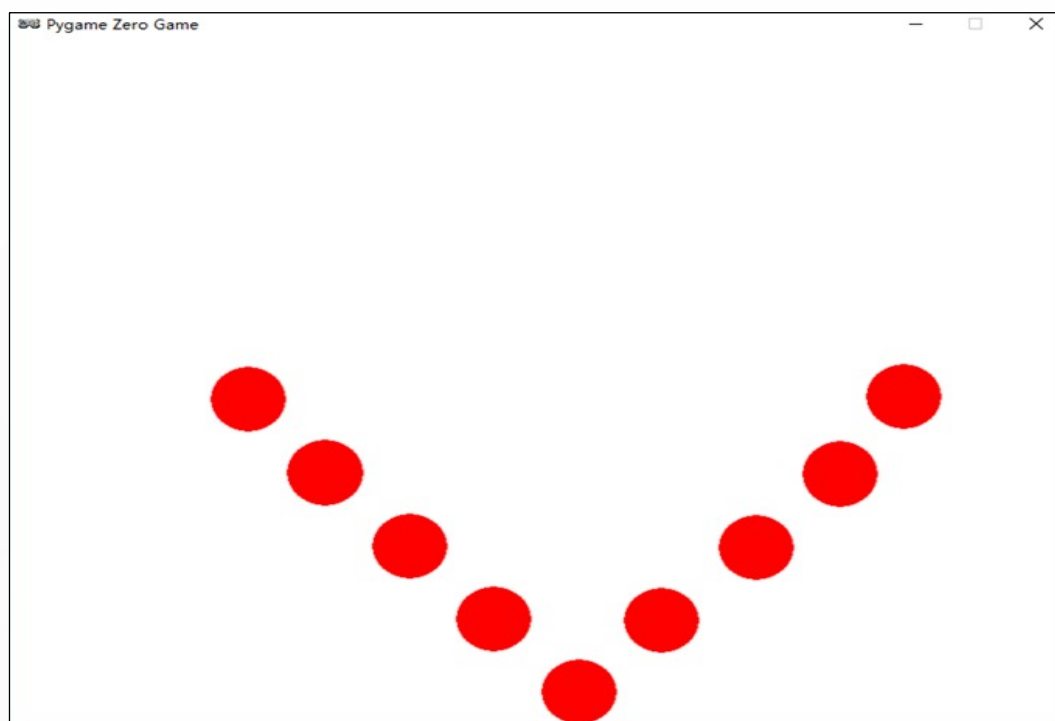
### 3-2-5.py

```
import pgzrun  
def draw():  
    screen.fill('white')  
    for r in range(10, 51, 10):  
        screen.draw.circle((400, 300), r, 'black')  
pgzrun.go()
```



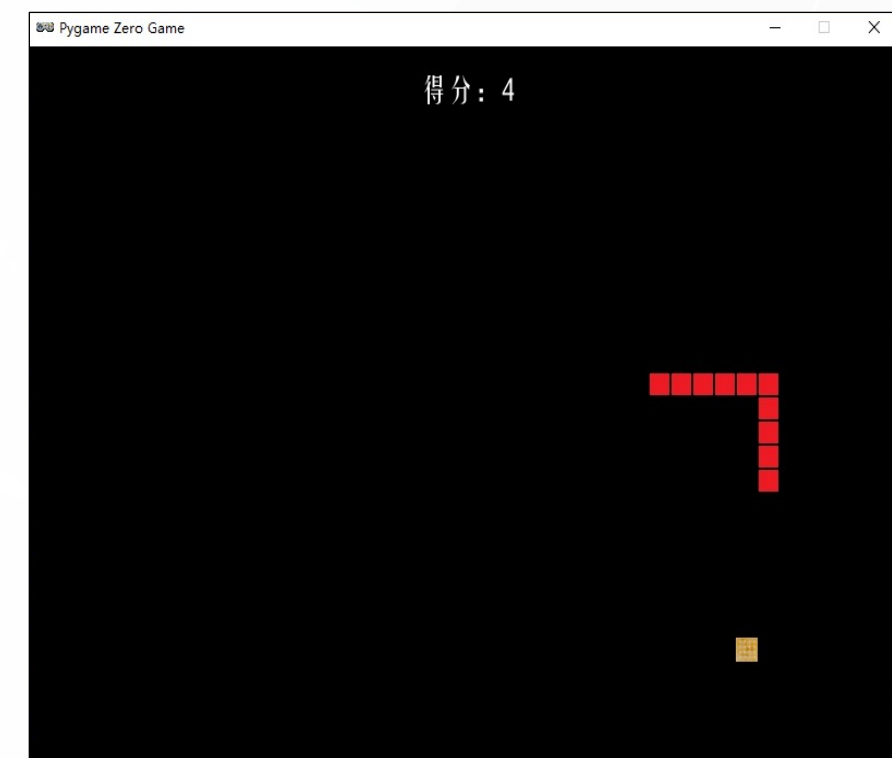
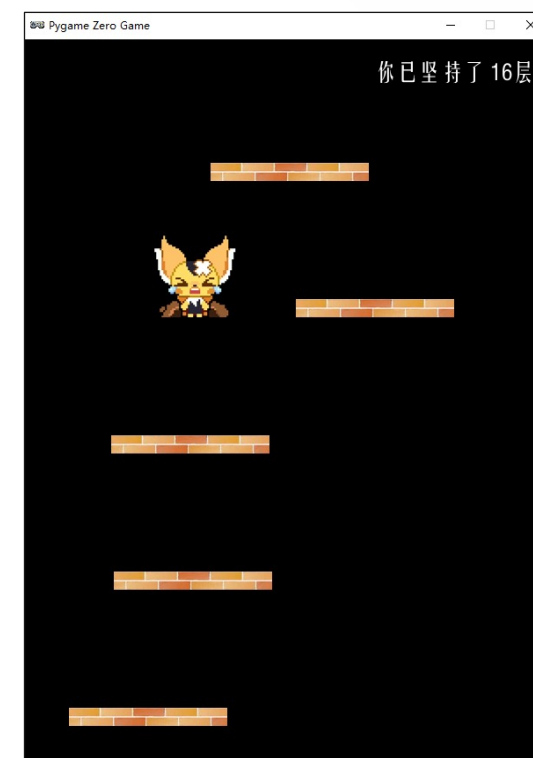
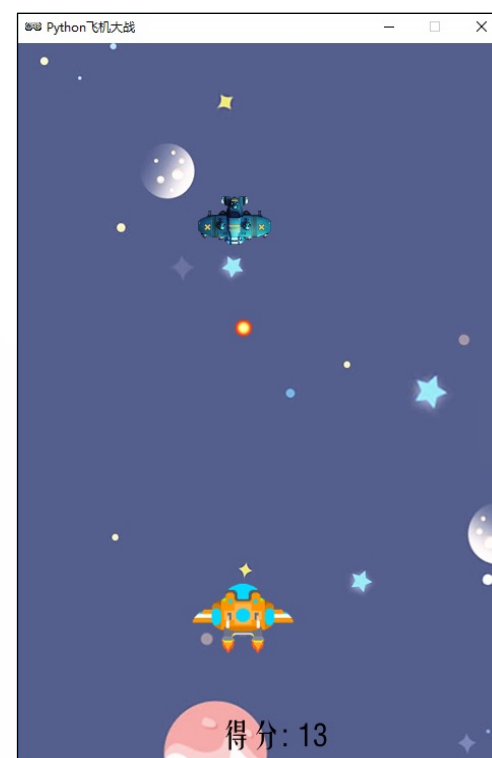
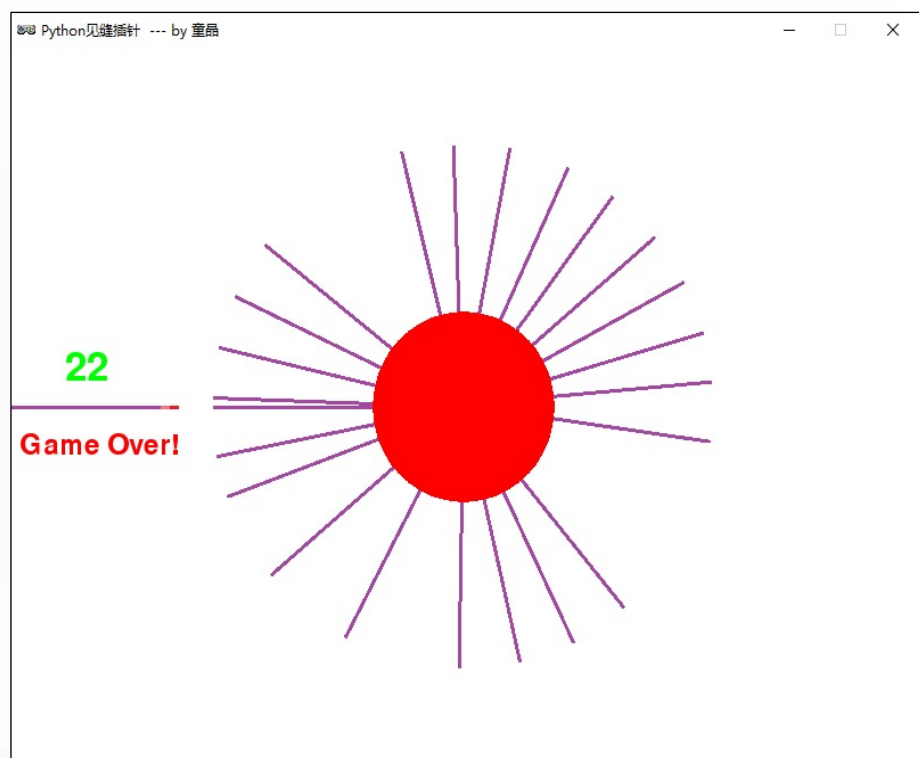
## 图书简介

- **第2章**学习字符串、变量、if语句等语法知识，实现了弹跳的小球程序（24行代码）
- **第3章**学习for循环语句、循环嵌套、随机数等语法知识，绘制了美丽的圆圈画（18行代码）
- **第4章**学习列表，并利用鼠标互动，实现了疯狂的小圆圈程序（38行代码）
- **第5章**学习数据类型转换、整数运算、图片导入与显示等功能，应用if语句，实现了飞翔的小鸟游戏（58行代码）



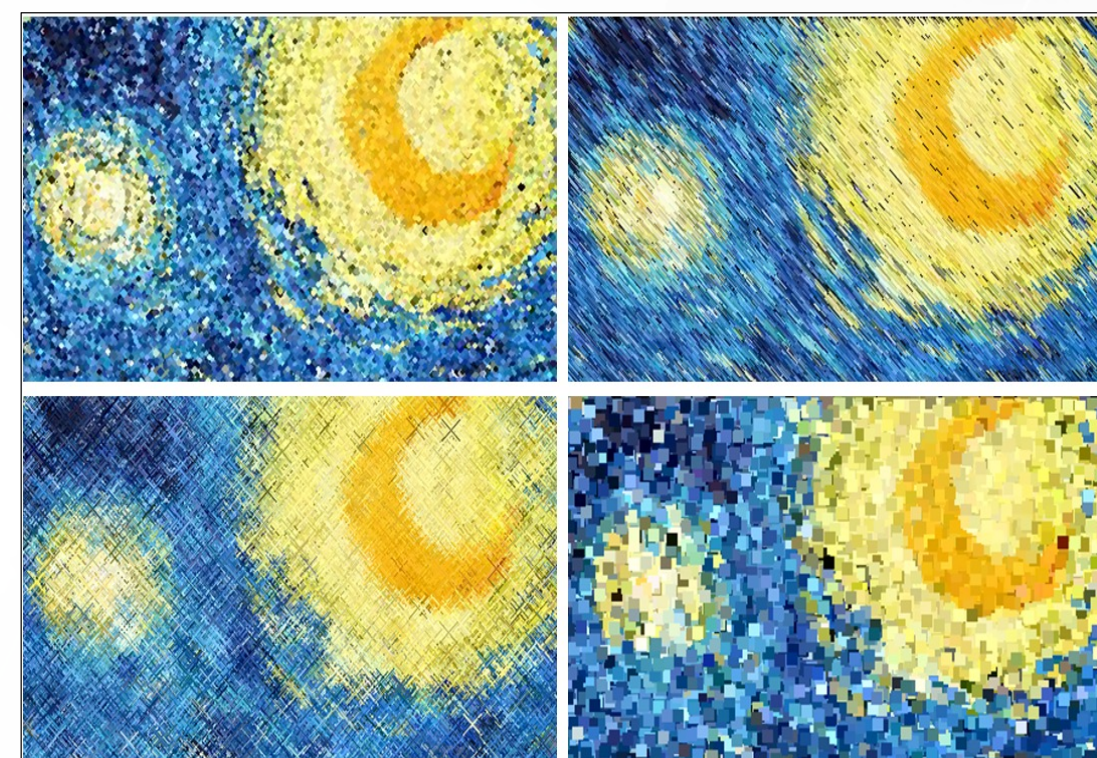
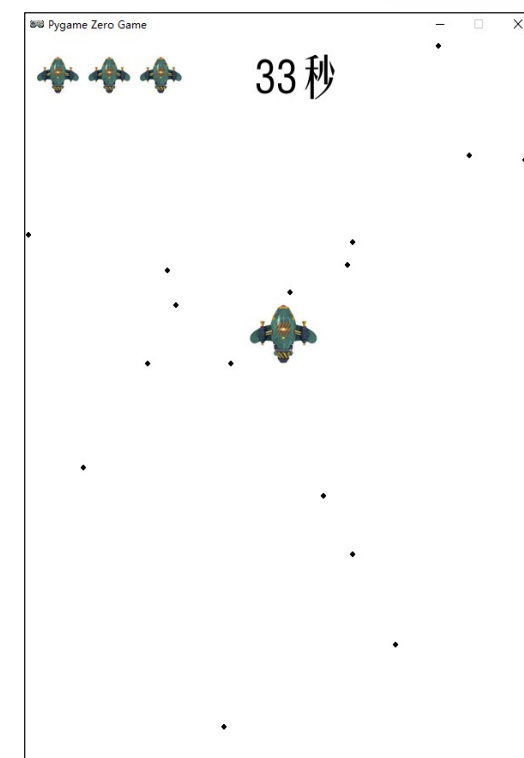
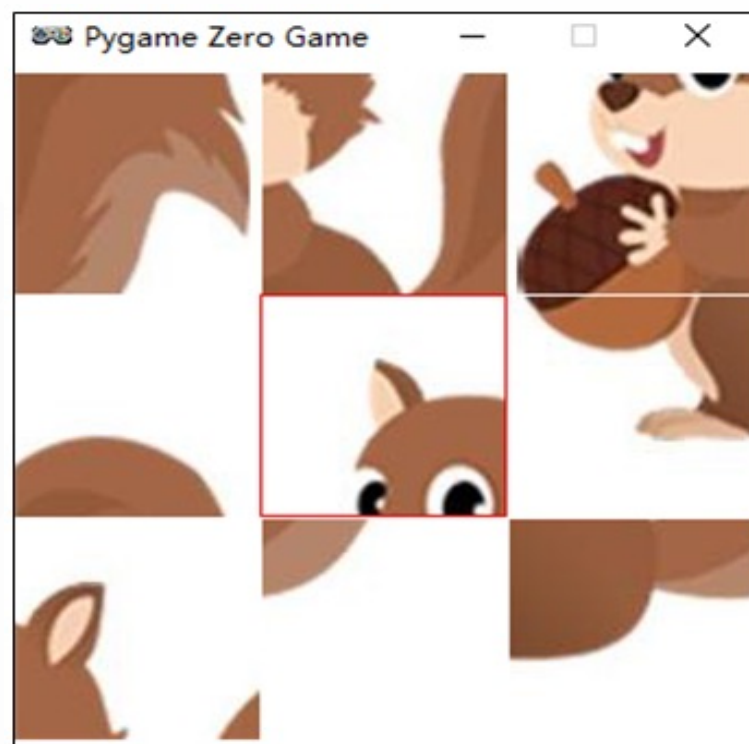
## 图书简介

- **第6章**学习if-elif-else和input语句、图片旋转、音效播放等功能，实现了见缝插针游戏（49行代码）
- **第7章**学习复合运算符、字符串拼接、布尔变量等语法知识，实现了飞机大战游戏（88行代码）
- **第8章**学习绝对值函数、列表元素删除等语法知识，应用键盘控制，实现了勇闯地下一百层游戏（74行代码）
- **第9章**学习列表元素插入、break与continue语句、函数的定义与调用等语法知识，实现了贪吃蛇游戏（97行代码）



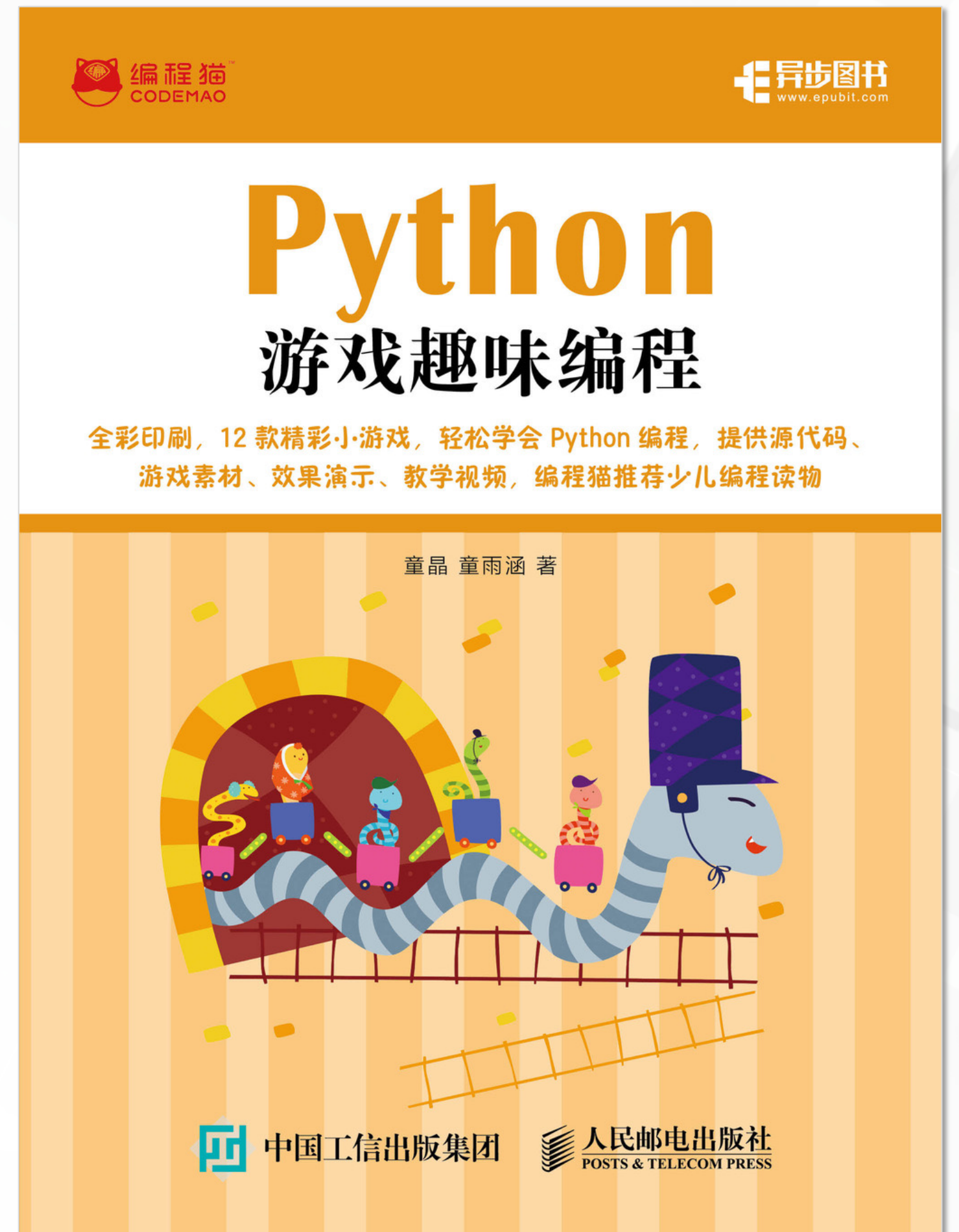
## 图书简介

- **第10章**学习时间模块、文件读写等语法知识，实现了拼图游戏（98行代码）
- **第11章**学习二维数组、元组、集合、字典、while循环等语法知识，实现了消灭星星游戏（98行代码）
- **第12章**学习面向对象编程，包括类和对象、成员变量、成员函数、构造函数、继承等语法知识，实现了坚持一百秒游戏（99行代码）
- **第13章**学习转义字符、异常处理等语法知识，利用第三方图像处理库实现了趣味图像的生成（70行代码）



## 图书简介

- 为初学者量身打造
  - 先讲较少的语法知识，利用这些语法知识编写趣味游戏，通过游戏案例逐步引入新的语法知识，便于读者学习理解
  - 案例从易到难，代码均不超过100行
- 趣味性强
  - 精选了12个游戏案例，学习有成就感
  - 分解步骤，即时反馈提升学习兴趣
- 提升可拓展性强
  - 多道练习题，附参考答案
  - 实现思路讲解、进一步实践方向



## 相关赛事介绍



钉钉扫码加入学习交流群

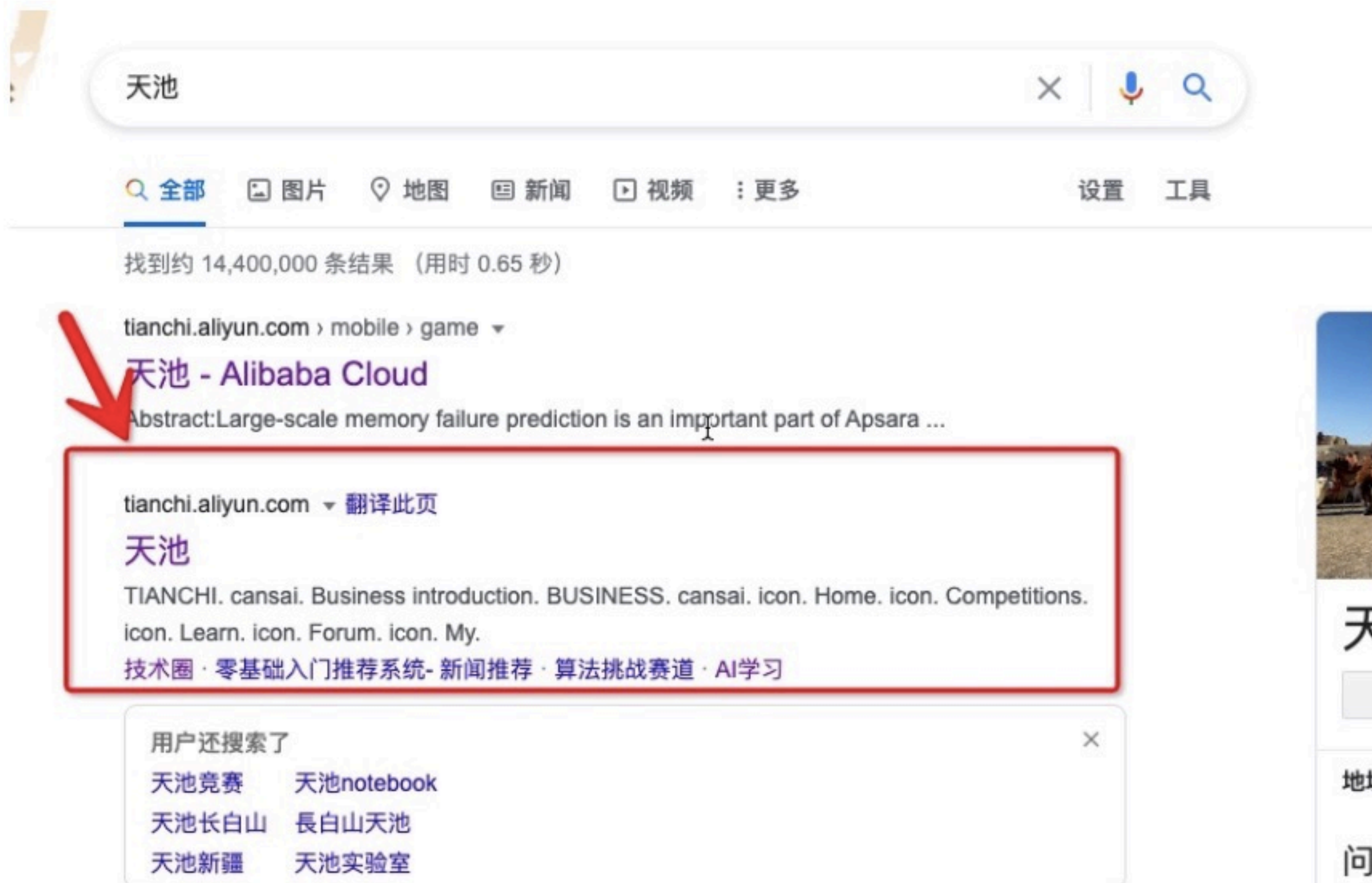
赛事地址：<https://tianchi.aliyun.com/aitalent>

这里给大家推荐下阿里云天池近期举办的“天池·TEENTOP杯”AI少年挑战赛，本次大赛分为Python能力闯关赛和数据科学挑战赛两个赛程，以**学+练+赛**的形式展开，配套有**入门训练营和进阶课程**，内容全面覆盖Python编程基础、数据分析、数据科学等内容，还有**优秀讲师亲自授课培训和专属钉钉群辅导**，青少年一起入门编程的学习好机会。

直播相关资料获取及回放查看地址：<https://tianchi.aliyun.com/specials/promotion/activity/bookclub>

# Q&A

1) 首先需要进入天池官网，大家打开浏览器，搜索 天池，找到 tianchi.aliyun.com即可访问进入天池官



2) 在天池官网，将鼠标移到 天池学习，即可出现下拉列表，点击 天池读书会，即可进入天池读书会的页面。





3) 在天池读书会页面，你可以对对应的读书会图书进行提问，优秀的提问还有机会获得赠书，还可以点击配套的训练营或者课程资源进入学习，还有点击实践代码获取读书会的项目实践的代码，跟着我一起进行项目实践和代码学习，同时还有很多其他的读书会，大家也可以观看举办过的读书会的回放，或者预约还没开始的读书会。



**童晶** 本书作者、浙大博士、河海大学副教授

直播主题 《Python游戏趣味编程》

直播时间 2021年6月23日 20:00

学习资料 Python训练营

实践项目 待定



[🗨️ 提问](#) | [📖 学习训练营](#) | [🛒 购买地址](#) | [📄 PPT下载](#) | [👉 实践代码](#) | [🕒 预约直播](#)

# 谢谢观看

