

# Python艺术编程08

## ——综合应用与游戏艺术

北京大学 陈斌

2018.11.05

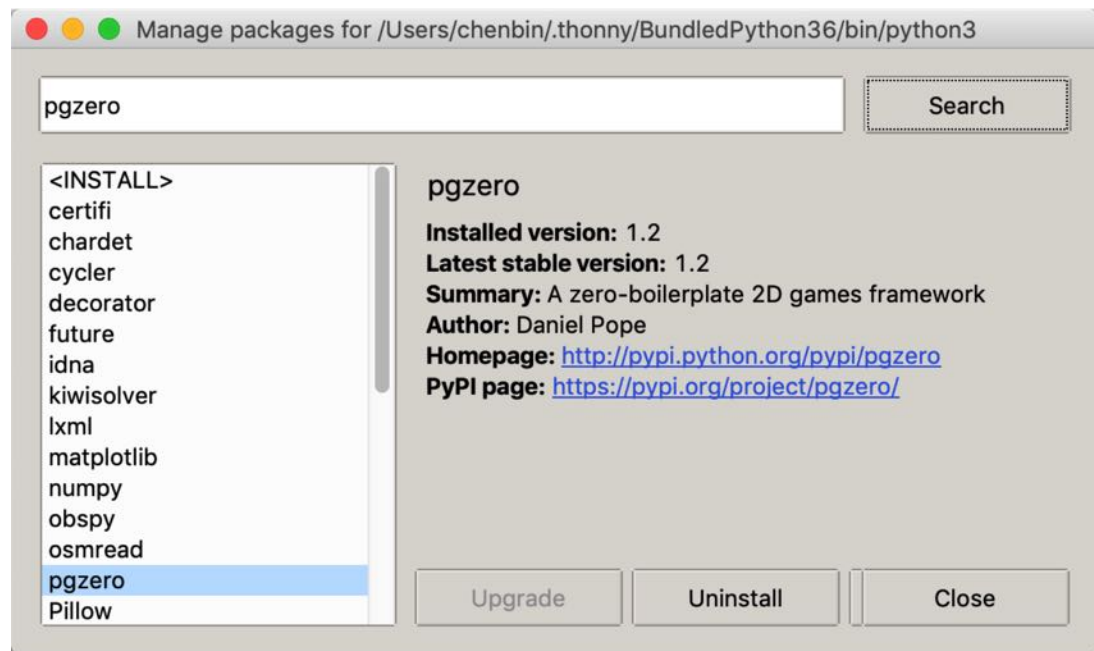
# 目录

- Pygame Zero基础
- 设计实现“接宝物”游戏



# Pygame Zero安装

- 打开Thonny
- 菜单“工具” -> “管理模块”
- 输入“pgzero”
- 点击搜索
- 点击“install”
- 安装成功!



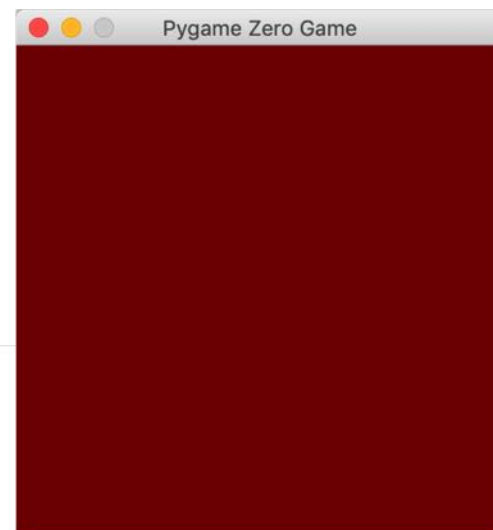


# 创建第一个窗口！

- 图形化的游戏都有一个窗口
- 我们来写程序创建一个300\*300的正方形红色窗口
  - 注意line1和15是必须的
- 注意！打开的窗口要按组合键CTRL-Q才能关闭
  - L6/7指定窗口的大小
  - L11/12是在刷新窗口的时候设置红色填充
- 每秒要调用draw函数60次！

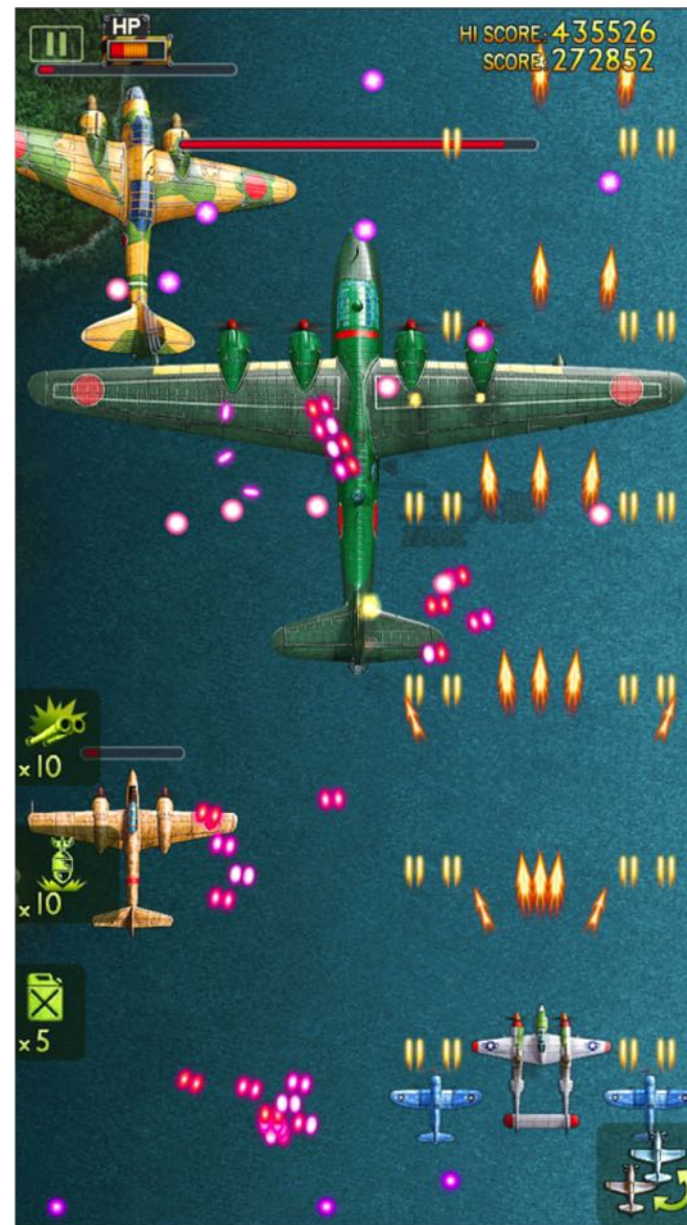


```
1 import pgzrun
2
3 # 绘制窗口背景为暗红色
4
5 # 窗口的宽和高设置
6 WIDTH = 300
7 HEIGHT = 300
8
9
10 # 每次需要刷新窗口的时候，会自动调用draw函数
11 def draw():
12     screen.fill((128, 0, 0))
13
14
15 pgzrun.go()
```



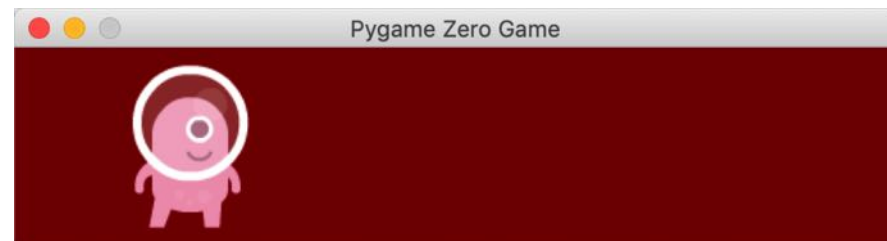
# 让游戏动起来

- 游戏中的图形分为：
  - 不动、或者虽然活动但没有影响的背景
  - 运动、或者虽然不动但会互相碰撞、互相作用的角色
  - 经常更新的各种信息显示
- 要让图形动起来，就要不停地做两件事
  - 更新：背景数据、角色位置
  - 重画：根据新的数据和位置重新绘制画面



# 创建游戏角色

- 在源代码文件目录下新建两个目录
  - images: 存放图片
  - sounds: 存放声音
- 用Actor类创建一个精灵对象
  - 指定图片、位置
- 在draw中画出精灵
  - 精灵图片将出现在指定的位置
  - alien.pos



```
1 import pgzrun
2
3 # 绘制一个精灵
4
5 # 1, 创建一个精灵
6 alien = Actor('alien')
7 alien.pos = 100, 56
8
9 # 2, 设定窗口大小
10 WIDTH = 500
11 HEIGHT = alien.height + 20
12
13
14 # 3, 每次需要刷新窗口的时候, 会自动调用draw函数
15 def draw():
16     screen.clear()
17     screen.fill((128,0,0))
18     alien.draw()
19
20
21 pgzrun.go()
```

images/alien.png

# 让精灵动起来

- 首先创建精灵alien
- 然后定义每帧的刷新需要做的两件事情
  - 记住每秒要刷新60次！
- 更新update
  - 改变精灵的位置，每秒会改变60次！
- 绘制draw
  - 填充窗口和画出精灵

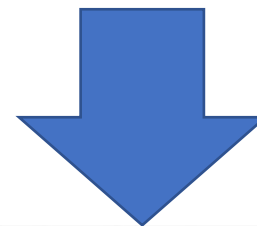
```
1  import pgzrun
2
3  # 让精灵运动起来
4  # 1, 创建一个精灵
5  alien = Actor('alien')
6  alien.topright = 0, 10
7
8  # 2, 设定窗口大小
9  WIDTH = 500
10 HEIGHT = alien.height + 20
11
12 # 3, 每次需要刷新窗口的时候, 会自动调用draw函数
13 def draw():
14     screen.clear()
15     screen.fill((128, 0, 0))
16     alien.draw()
17
18 # 4, 每一帧都会自动调用update函数
19 def update():
20     alien.left += 2
21     if alien.left > WIDTH:
22         alien.left = 0
23
24
25 pgzrun.go()
```

# 与精灵交互

## 用鼠标点击



pos就是鼠标的位置



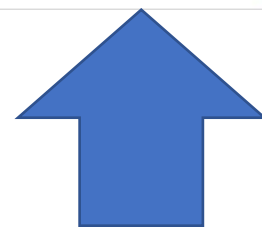
```
24 # 5, 当鼠标点击当时调用这个函数  
25 def on_mouse_down(pos):  
26     if alien.collidepoint(pos):  
27         print("Ekk")  
28     else:  
29         print("You missed me!")  
30
```



# 还是用图形和声音来响应更好

- 当鼠标点击时
  - 测试鼠标位置是否与外星人角色重叠
  - 播放音效
  - 外星人形象切换为hurt
- 有个小毛病
  - 切换是一次性的
  - 没有恢复成正常的alien
- 修改方案让外星人1秒后恢复?
  - sleep?

```
27 # 5, 当鼠标点击当时调用这个函数
28 # 切换外星人形象, 播放音效
29 def on_mouse_down(pos):
30     if alien.collidepoint(pos):
31         sounds.eep.play()
32         alien.image = 'alien_hurt'
```



images/alien.png  
images/alien\_hurt.png  
sounds/eep.wav

# 定时器clock

- 定时器clock可以安排在一段时间后自动调用某个函数
  - `clock.schedule_unique()`
- 将外星人受伤和恢复变为两个函数
  - `set_alien_hurt`
  - `set_alien_normal`
- 鼠标点击时调用受伤函数
- 在受伤函数里安排1s后恢复

```
26 # 5, 当鼠标点击当时调用这个函数
27 # 切换外星人形象, 播放音效
28 def on_mouse_down(pos):
29     if alien.collidepoint(pos):
30         set_alien_hurt()
31
32 # 6, 受伤外星人
33 def set_alien_hurt():
34     alien.image = 'alien_hurt'
35     sounds.eep.play()
36     # 设置定时器, 1s后自动恢复
37     clock.schedule_unique(set_alien_normal, 1.0)
38
39 # 7, 外星人恢复
40 def set_alien_normal():
41     alien.image = 'alien'
```

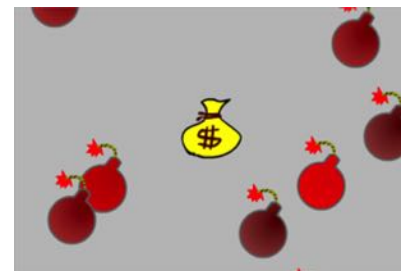
# 大功告成！

- 设定窗口大小WIDTH/HEIGHT
- 创建一批角色Actor
- 写好每次刷新都要做的事情
  - 更新：修改背景和角色的数据
  - 重画：画一遍窗口的内容
- 写好交互的动作
  - 鼠标：on\_mouse\_down(pos)
  - 键盘：on\_key\_down(key)



# 写一个接宝物的游戏

- 角色确定
  - 哪几种类型角色？
  - 角色的运动？控制？
  - 角色之间的碰撞？
- 画面设计
  - 背景画面、显示信息
  - 角色的绘制位置
- 交互设计
  - 鼠标控制？





# 一些进一步的参考

- 颜色可以是(r,g,b)也可以是字符串名称
  - (128,0,0)或者'red'
- 窗口写字：颜色可以是(r,g,b)也可以是字符串名称
  - `screen.draw.text(str, (x,y), color=颜色, background=颜色)`
- 绘制背景图像
  - `screen.clear()`
  - `screen.fill(颜色)`
  - `screen.blit(背景图名, (x,y))`
- 音效sounds
  - `sounds.<名称>.play(loops=<重复次数>)`
  - `sounds.<名称>.stop()`

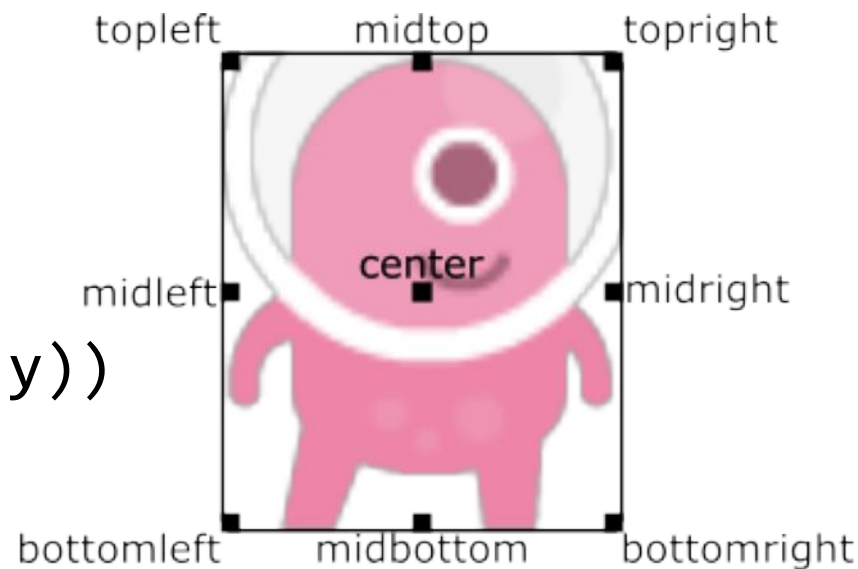
# 一些进一步的参考

- 定时器clock

- 安排时间: `clock.schedule(<函数>, <延迟时间>)`
- 会覆盖以前的安排: `clock.schedule_unique(<函数>, <延迟时间>)`
- 定期反复调用: `clock.schedule_interval(<函数>, <重复时间>)`
- 取消安排: `clock.unschedule(<函数>)`

- 角色Actor

- 位置: `center=(x,y)`
- 角度旋转: `angle=角度值`
- 相对距离: `distance_to(<另一个角色>/(x,y))`
- 相对角度: `angle_to(<另一个角色>/(x,y))`



# 一些进一步的参考

- 角色Actor
  - image=图片
- Actor碰撞检测
  - collidepoint(<另一个角色>/(x,y))
  - colliderect(<另一个角色>/<矩形Rect()>/((x1,y1),(x2,y2)))
- 动画效果
  - animate(<角色>, pos=(x,y))
- 合成的音符
  - tone.play('E4', 0.5)
  - 创建一个，后续调用play: beep = tone.create('A3', 0.5)
  - beep.play()

# 一些进一步的参考

- 鼠标事件

- 鼠标左右键: `mouse.LEFT`, `mouse.RIGHT`, `mouse.MIDDLE`
- `on_mouse_move(pos, rel, buttons)`
  - if `mouse.LEFT` in `buttons`:
- `on_mouse_down(pos, button)`

- 键盘事件

- `on_key_down(key)`
- 键名称: `key.A`, `key.B`, `key.ESC`...