程序员式浪漫：Python 带你看雪啦！

原创

穿背心儿的程序猿 于 2018-12-26 14:45:25 发布 1932 收藏 1

作者 | Ahab

责编 | 仲培艺

前段时间笔者写了一篇题为《用 Python 来一场人工造雪》的文章，但大家似乎都不满足仅仅是一个图片的雪花，都想来一场动态的人工降雪。于是便有了下面的内容：

动态视频链接：https://v.qq.com/x/page/q1354od8qni.html?start=12

一、具体介绍

动态版的实现主要是依靠 pygame 这个模块，从绘制到实现动态移动，最初的想法是按照飞机大战的思路把飞机替换成上次绘制的科赫雪花，自己试了试效果很差就是一张张图片在屏幕上乱飞，通过百度发现 pygame.draw 模块，跟 turtle 差不多。当然还是少不了 random，自己做过的 Python 面试题【BAT版】（02）中 random 算是常客，学习的时候觉得就是一个随机数生成而已，最近慢慢接触才发现 randmo 的强大，怪不得成为公司面试的必考题。

二、代码实现

因为使用 pygame 第一步要做的就是初始化：

```
import pygame
import random
#初始化
pygame.init()
```

加载背景图同时根据背景图的大小设置屏幕长宽：

```
SIZE = (1000, 500)
screen = pygame.display.set_mode(SIZE)
pygame.display.set_caption("下雪了")
#加载位图
background = pygame.image.load('snow.jpg')
```

接下来我们要定义一个雪花列表，且初始化雪花，这里需要使用 random 随机数设置 xy 轴的坐标和速度。

random.randrange

random.randrange([start],stop[, step])：从指定范围内，按指定基数递增的集合中获取一个随机数。

random.randint(a,b)：用于生成一个指定范围内的整数。其中参数a是下限，参数b是上限，生成的随机数n：a<=n<=b。

```
# 定义一个雪花列表
snow = []
# 初始化雪花
for i in range(300):
x = random.randrange(0, SIZE[0])
y = random.randrange(0, SIZE[1])
speedx = random.randint(-1, 2)
speedy = random.randint(3,8)
snow.append([x, y, speedx, speedy])
```

做过飞机大战或者熟悉pygame的朋友应该知道接下来要做的就是设置游戏循环，同时还将之前加载的背景图进行了绘制。

Surface对象有一个名为blit（）的方法，它可以绘制位图

screen.blit(space, (0,0))

第一个参数是加载完成的位图，第二个参数是绘制的起始坐标。

```
done = False
while not done:
# 消息事件循环，判断退出
for event in pygame.event.get():
if event.type == pygame.QUIT:
done = True
#绘制位图
screen.blit(background, (0,0))
```

这一步是最重要的绘制雪花和设置雪花列表的循环。

绘制雪花使用的是pygame.draw 模块，此模块用于在Surface上绘制一些简单的图形，比如点、直线、矩形、圆、弧等。我们绘制雪花用到的是：

pygame.draw.circle

原型：pygame.draw.circle(Surface, color, pos, radius, width=0): return Rect

用途：用于绘制圆形。第三个参数pos是圆心的位置坐标，radius指定了圆的半径。

width参数表示线条（画笔）的宽度，如果该值设置为0，则表示填充整个图形，我们的绘制的雪花就是使用填充白色。颜色参数通常是一个RGB三元组（R, G, B）。

雪花列表循环主要取决于雪花列表的长度，同样设置了移动雪花的位置，程序中还做了一个判定雪花从顶端向下移动时如果落出屏幕，将会重设位置。

```
# 雪花列表循环
for i in range(len(snow)):
# 绘制雪花，颜色、位置、大小
pygame.draw.circle(screen, (255, 255, 255), snow[i][:2], snow[i][3])
```

5.

```
# 移动雪花位置（下一次循环起效）
snow[i][0] += snow[i][2]
snow[i][1] += snow[i][3]
```

9.

```
# 如果雪花落出屏幕，重设位置
if snow[i][1] > SIZE[1]:
snow[i][1] = random.randrange(-50, -10)
snow[i][0] = random.randrange(0, SIZE[0])
```

到这程序基本就写完了，只需要添加刷新屏幕的时间和游戏退出语句就完事了。

```
pygame.display.flip()
clock.tick(20)
```

3.

```
pygame.quit()
```

最后感谢【唐僧不爱八戒】提供 pygame.draw 这个思路