

# Python+Pygame 游戏学习

## （初级教程）

小五义

<http://www.cnblogs.com/xiaowuyi>

本文档通过博客整理而成，所用素材  
请从

[http://www.kuaipan.cn/file/id\\_749007936687911.html](http://www.kuaipan.cn/file/id_749007936687911.html) 下载。

# 第一章：安装及矩形、圆型画图

pygame 是一个设计用来开发游戏的 python 模块，其实说白了和 time、os、sys 都是一样的东东。今天开始正式学习 pygame，下载地址：[www.pygame.org](http://www.pygame.org)，具体安装方法就不讲了，很简单。下载后安装完成即可，在 pygame 的学习中，本文使用了 spe 编辑器，感觉还不错。

## 1、pygame 窗口

pygame 绘制图形前，首先需要建立一个窗口，说来很简单，请看下面的代码，怎么样，是不是很简单。

```
import pygame #这句不用注释了吧，呵呵
```

```
pygame.init() #模块初始化，任何 pygame 程序均需要执行此句
```

```
screen=pygame.display.set_caption('hello world')#定义窗口的标题为'hello world'
```

```
screen=pygame.display.set_mode([640,480]) #定义窗口大小为640*480
```

```
screen.fill([255,255,255])#用白色填充窗口
```

## 2、窗口退出

pygame 有一个事件循环，不断检查用户在做什么。事件循环中，如何让循环中断下来（pygame 形成的窗口中右边的插号在未定义前是不起作用的），常用的代码如下：

```
while True:
```

```
    for event in pygame.event.get():
```

```
        if event.type==pygame.QUIT:
```

```
            sys.exit()
```

## 3、pygame 中的颜色

在 screen.fill([255,255,255])这一语句中，已经看出，pygame 使用的是 RGB 系统。纯绿色用[0,255,0]，纯蓝色用[0,0, 255]，纯红色用[255,0,0]。如果不使用 RGB 记法，pygame 还提供了一个命名颜色列表，也可以直接使用这些命名颜色。定义好的颜色句有600多个，可以在 colordict.py 文件中查看具体名称。使用命名列表时，首先要在程序最前面导入 THECOLORS。

```
from pygame.color import THECOLORS
```

然后使用某个命名颜色：

```
pygame.draw.circle(screen,THECOLORS["red"],[100,100],30,0)
```

## 4、圆形

pygame.draw.circle()用来画圆形，具体包括五个参数：(1)画圆的表面，在本例中用 screen 创建了一个窗口，所以是画在 screen 表面上。(2)用什么颜色来画，如用红色[255,0,0]。(3)在什么位置画，[top,left]。(4)直径。(5)线宽，其中0表示完成填充。

```
pygame.draw.circle(screen,[255,0,0],[100,100],30,0)
```

## 5、矩形

pygame.draw.rect()用来创建一个矩形。Rect(left,top,width,height)用来定义位置和宽高，具体代码如下：

```
pygame.draw.rect(screen,[255,0,0],[250,150,300,200],0)
```

也可以用下面的定义方法

```
rect_list=[250,150,300,200]
```

```
pygame.draw.rect(screen,[255,0,0],rect_list,0)
```

或者

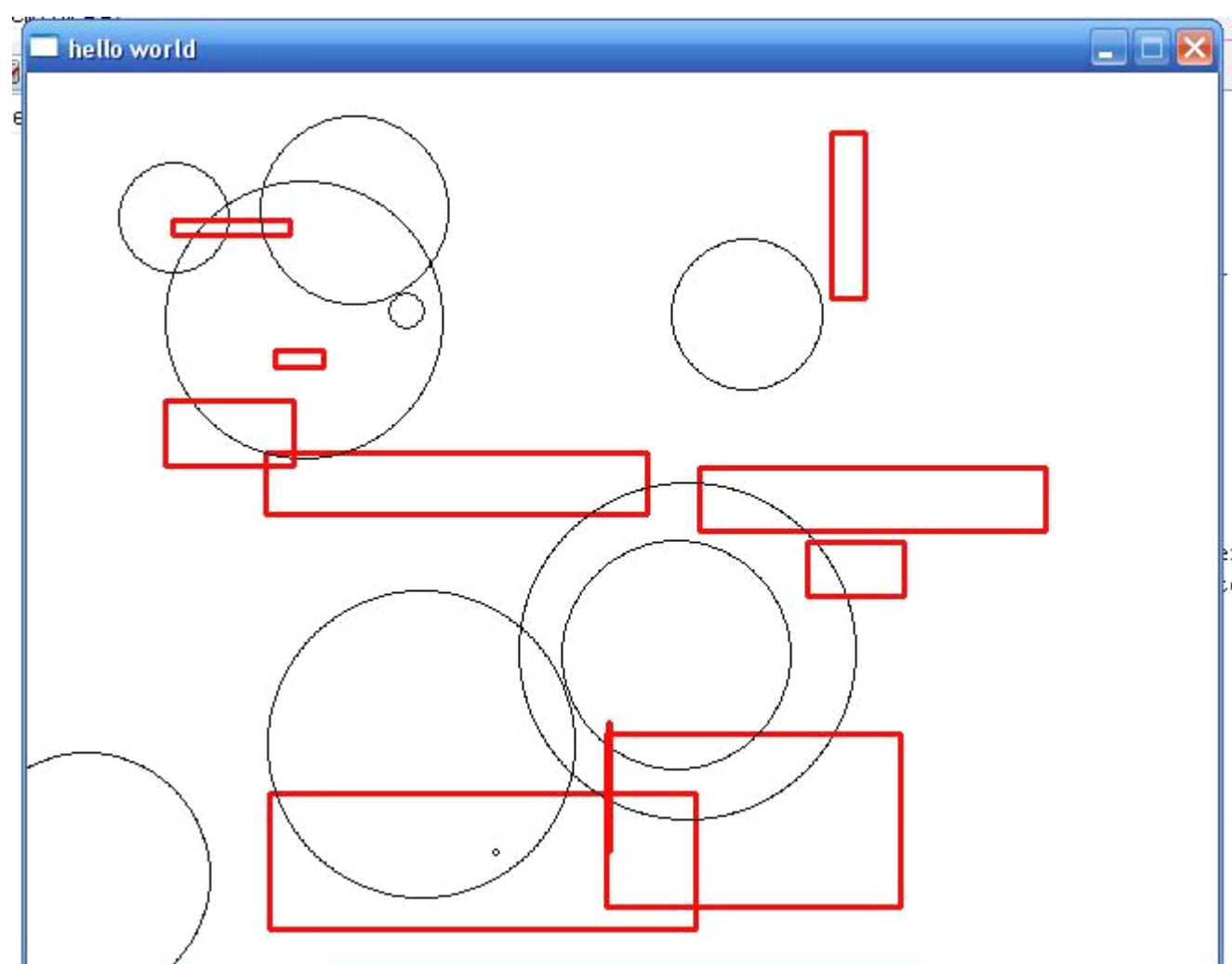
```
my_rect=pygame.Rect(250,150,300,200)
pygame.draw.rect(screen,[255,0,0],my_rect,0)
```

## 6、实例

利用 random 模块随机生成大小和位置在表面上绘画，具体代码如下：

#@小五义 <http://www.cnblogs.com/xiaowuyi>

```
import pygame,sys
import time
import random
pygame.init()
screencaption=pygame.display.set_caption('hello world')
screen=pygame.display.set_mode([640,480])
screen.fill([255,255,255])
for i in range(10):
    zhijing=random.randint(0,100)
    width=random.randint(0,255)
    height=random.randint(0,100)
    top=random.randint(0,400)
    left=random.randint(0,500)
    pygame.draw.circle(screen,[0,0,0],[top,left],zhijing,1)
    pygame.draw.rect(screen,[255,0,0],[left,top,width,height],3)
pygame.display.flip()
while True:
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            sys.exit()
```



## 第二章：从画点到动画

### 1、单个像素（画点）

利用 pygame 画点主要有三种方法：

方法一：画长宽为1个像素的正方形

#@小五义 <http://www.cnblogs.com/xiaowuyi>

```
import pygame,sys
pygame.init()
screen=pygame.display.set_caption('hello world!')
screen=pygame.display.set_mode([640,480])
screen.fill([255,255,255])
pygame.draw.rect(screen,[0,0,0],[150,50,1,1],1) #画1*1的矩形，线宽为1，这里不能是0，因为1*1无空白区域。
pygame.display.flip()
while True:
```

```
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            sys.exit()
```

方法二：画个直径为1的圆

#@小五义 <http://www.cnblogs.com/xiaowuyi>

```
import pygame,sys
pygame.init()
screen=pygame.display.set_caption('hello world!')
screen=pygame.display.set_mode([640,480])
screen.fill([255,255,255])
pygame.draw.circle(screen,[0,0,0],[150,200],1,1)
pygame.display.flip()
while True:
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            sys.exit()
```

方法三：这种方法并不是画上去的，而是改变了 **surface** 上某个点的颜色，这样看上去像是画了一个点 **screen.set\_at()**。另外，如果要得到某个像素的颜色，可以使用 **screen.get\_at()**。

#@小五义 <http://www.cnblogs.com/xiaowuyi>

```
import pygame,sys
pygame.init()
screen=pygame.display.set_caption('hello world!')
screen=pygame.display.set_mode([640,480])
screen.fill([255,255,255])
screen.set_at([150,150],[255,0,0])#将150, 150改为红色。
pygame.display.flip()
while True:
    for event in pygame.event.get():
```

```
if event.type==pygame.QUIT:
    sys.exit()
```

## 2、连接多个点形成线

`pygame.draw.lines()`方法可以将多个点连接成为线。该方法有5个参数：`surface` 表面、颜色、闭合线或者非闭合线（如果闭合 为 `True`, 否则为 `False`），点的列表，线宽。`pygame.draw.lines(surface, [color], False/True, plotpoints, 1)`。下面的例子画出了一条马路，具体如下：

#@小五义 <http://www.cnblogs.com/xiaowuyi>

```
import pygame, sys
def lineleft(): #画马路左边界
    plotpoints=[]
    for x in range(0,640):
        y=-5*x+1000plotpoints.append([x,y])
        pygame.draw.lines(screen,[0,0,0],False,plotpoints,5)
    pygame.display.flip()
def lineright():#画马路右边界
    plotpoints=[]
    for x in range(0,640):
        y=5*x-2000plotpoints.append([x,y])
        pygame.draw.lines(screen,[0,0,0],False,plotpoints,5)
    pygame.display.flip()
def linemiddle():#画马路中间虚线
    plotpoints=[]
    x=300
    for y in range(0,480,20):
        plotpoints.append([x,y])
        if len(plotpoints)==2:
            pygame.draw.lines(screen,[0,0,0],False,plotpoints,5)
            plotpoints=[]
    pygame.display.flip()

pygame.init()
screen=pygame.display.set_caption('hello world!')
screen=pygame.display.set_mode([640,480])
screen.fill([255,255,255])
lineleft()
lineright()
linemiddle()
while True:
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            sys.exit()
```

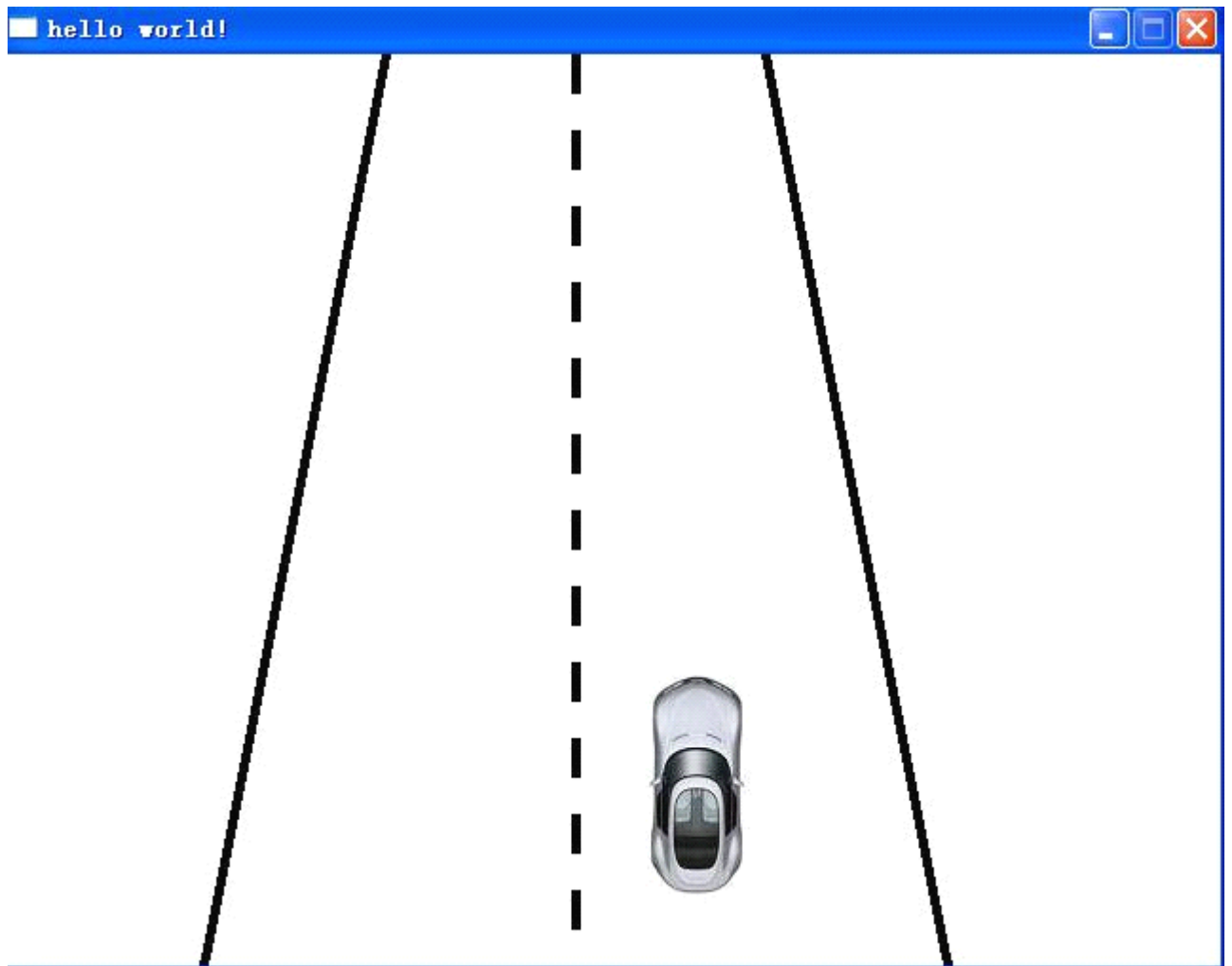
## 3、引用图像

在 pygame 中引用图像最简单的以夷伐夷是 image 函数。下面在马路的实例中，加入一辆汽车。首先 pygame.image.load()函数从硬盘 加载一个图像，并创建一个名为 my\_car 的对象。这里，my\_car 是一个 surface，不过是存在内存中，并未显示出来，然后用 blit（块移）方法将 my\_car 复制到 screen 表面上，从而显示出来。具体代码如下：

#@小五义 <http://www.cnblogs.com/xiaowuyi>

```
import pygame,sys
def lineleft():
    plotpoints=[]
    for x in range(0,640):
        y=-5*x+1000plotpoints.append([x,y])
        pygame.draw.lines(screen,[0,0,0],False,plotpoints,5)
    pygame.display.flip()
def lineright():
    plotpoints=[]
    for x in range(0,640):
        y=5*x-2000plotpoints.append([x,y])
        pygame.draw.lines(screen,[0,0,0],False,plotpoints,5)
    pygame.display.flip()
def linemiddle():
    plotpoints=[]
    x=300
    for y in range(0,480,20):
        plotpoints.append([x,y])
        if len(plotpoints)==2:
            pygame.draw.lines(screen,[0,0,0],False,plotpoints,5)
            plotpoints=[]
    pygame.display.flip()
def loadcar(): #载入 car 图像
    my_car=pygame.image.load('ok1.jpg') #当前文件夹下的 ok1.jpg 文件
    screen.blit(my_car,[320,320])
    pygame.display.flip()

pygame.init()
screen=pygame.display.set_caption('hello world!')
screen=pygame.display.set_mode([640,480])
screen.fill([255,255,255])
lineleft()
lineright()
linemiddle()
loadcar()
while True:
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            sys.exit()
```



素材: ok1.jpg



#### 4、动画

计算机动画实际上就是把图像从一个地方移动到另一个地方，同时几个连续动作交待显示就会产生逼真的效果。因此，在做动画中，最基本要考虑的因素主要是三个，一是时间，什么时间移动，多长时间变下一个动作，二是位置，从什么位置到什么位置，三是动作，前后两个动作的连续性。在这个例子中，因为车是俯视的，所以车轮转动实际是看不到的，所以不用考虑连续动作的变化，而是只考虑车的位置和多长时间移动即可。第一步 `pygame.time.delay()` 来实现 时间延迟；第二步利用 `pygame.draw.rect()` 把原来位置的图像覆盖掉；第三步 `screen.blit()` 在新位置引入图像。下面的例子实现了汽车从驶入到驶出的过程。

#@小五义 <http://www.cnblogs.com/xiaowuyi>



```

import pygame,sys
def lineleft():
    plotpoints=[]
    for x in range(0,640):
        y=-5*x+1000        plotpoints.append([x,y])
    pygame.draw.lines(screen,[0,0,0],False,plotpoints,5)
    pygame.display.flip()
def lineright():
    plotpoints=[]
    for x in range(0,640):
        y=5*x-2000        plotpoints.append([x,y])
    pygame.draw.lines(screen,[0,0,0],False,plotpoints,5)
    pygame.display.flip()
def linemiddle():
    plotpoints=[]
    x=300
    for y in range(0,480,20):
        plotpoints.append([x,y])
        if len(plotpoints)==2:
            pygame.draw.lines(screen,[0,0,0],False,plotpoints,5)
            plotpoints=[]
    pygame.display.flip()
def loadcar(yloc):
    my_car=pygame.image.load('ok1.jpg')
    locationxy=[310,yloc]
    screen.blit(my_car,locationxy)
    pygame.display.flip()

if __name__=='__main__':
    pygame.init()
    screen=pygame.display.set_caption('hello world!')
    screen=pygame.display.set_mode([640,480])
    screen.fill([255,255,255])
    lineleft()
    lineright()
    linemiddle()

    while True:
        for event in pygame.event.get():
            if event.type==pygame.QUIT:
                sys.exit()
            for loop in range(480,-140,-50):
                pygame.time.delay(200)

```

```
pygame.draw.rect(screen,[255,255,255],[310,(looper+132),83,132],0)  
loadcar(looper)
```



## 第三章：时间、事件、文字

### 1、运动速率

上节中，实现了一辆汽车在马路上由下到上行驶，并使用了 `pygame.time.delay(200)` 来进行时间延迟。看了很多参考材料，基本每个材料都会谈到不同配置机器下运动速率的问题，有的是通过设定频率解决，有的是通过设定速度解决，自己本身水平有限，看了几篇，觉得还是《Beginning Game Development with Python and Pygame》这里面提到一个方法比较好。代码如下，代码里更改的地方主要是 `main` 里的代码，其中利用 `clock=pygame.time.Clock()` 来定义时钟，`speed=250.0` 定义了速度，每秒250像素，`time_passed=clock.tick()` 为上次运行时间单位是毫秒，`time_passed_seconds=time_passed/1000.0` 将单位改为秒，`distance_moved=time_passed_seconds*speed` 时间乘以速度得到移动距离，这样就能保证更加流畅。

#@小五义 <http://www.cnblogs.com/xiaowuyi>

```
import pygame,sys
def lineleft():
    plotpoints=[]
    for x in range(0,640):
        y=-5*x+1000        plotpoints.append([x,y])
    pygame.draw.lines(screen,[0,0,0],False,plotpoints,5)
    pygame.display.flip()
def lineright():
    plotpoints=[]
    for x in range(0,640):
        y=5*x-2000        plotpoints.append([x,y])
    pygame.draw.lines(screen,[0,0,0],False,plotpoints,5)
    pygame.display.flip()
def linemiddle():
    plotpoints=[]
    x=300
    for y in range(0,480,20):
        plotpoints.append([x,y])
        if len(plotpoints)==2:
            pygame.draw.lines(screen,[0,0,0],False,plotpoints,5)
            plotpoints=[]
    pygame.display.flip()
def loadcar(yloc):
    my_car=pygame.image.load('ok1.jpg')
    locationxy=[310,yloc]
    screen.blit(my_car,locationxy)
    pygame.display.flip()
```

```
if __name__=='__main__':
```

```

pygame.init()
screen=pygame.display.set_caption('hello world!')
screen=pygame.display.set_mode([640,480])
screen.fill([255,255,255])
lineleft()
lineright()
linemiddle()

clock=pygame.time.Clock()
looper=480    speed=250.0
while True:
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            sys.exit()

    pygame.draw.rect(screen,[255,255,255],[310,(looper+132),83,132],0)
    time_passed=clock.tick()
    time_passed_seconds=time_passed/1000.0
    distance_moved=time_passed_seconds*speed
    looper-=distance_moved

    if looper<=-480:
        looper=480
    loadcar(looper)

```

## 2、事件

我理解的就是用来解决键盘、鼠标、遥控器等输入后做出什么反映的。例如上面的例子，可以通过按上方向键里向上来使得小车向上移动，按下向下，使得小车向下 移动。当小车从下面倒出时，会从上面再出现，当小车从上面驶出时，会从下面再出现。代码如下。  
`event.type == pygame.KEYDOWN` 用来定义事件类型，`if event.key==pygame.K_UP` 这里是指当按下向上箭头时，车前进。`if event.key==pygame.K_DOWN` 则相反，指按下向下箭头，车后退。

#@小五义 <http://www.cnblogs.com/xiaowuyi>

```

import pygame,sys
def lineleft():
    plotpoints=[]
    for x in range(0,640):
        y=-5*x+1000        plotpoints.append([x,y])
    pygame.draw.lines(screen,[0,0,0],False,plotpoints,5)
    pygame.display.flip()
def lineright():
    plotpoints=[]
    for x in range(0,640):
        y=5*x-2000        plotpoints.append([x,y])

```

```

pygame.draw.lines(screen,[0,0,0],False,plotpoints,5)
pygame.display.flip()
def linemiddle():
    plotpoints=[]
    x=300
    for y in range(0,480,20):
        plotpoints.append([x,y])
        if len(plotpoints)==2:
            pygame.draw.lines(screen,[0,0,0],False,plotpoints,5)
            plotpoints=[]
    pygame.display.flip()
def loadcar(yloc):
    my_car=pygame.image.load('ok1.jpg')
    locationxy=[310,yloc]
    screen.blit(my_car,locationxy)
    pygame.display.flip()

if __name__=='__main__':
    pygame.init()
    screen=pygame.display.set_caption('hello world!')
    screen=pygame.display.set_mode([640,480])
    screen.fill([255,255,255])
    lineleft()
    lineright()
    linemiddle()

    loop=480

    while True:
        for event in pygame.event.get():
            if event.type==pygame.QUIT:
                sys.exit()
            elif event.type == pygame.KEYDOWN:

                if event.key==pygame.K_UP:
                    loop=loop-50
                    if loop<-480:
                        loop=480
                    pygame.draw.rect(screen,[255,255,255],[310,(loop+132),83,132],0)
                    loadcar(loop)
                if event.key==pygame.K_DOWN:
                    loop=loop+50
                    if loop>480:

```

```

        loopер=-480
pygame.draw.rect(screen,[255,255,255],[310,(loopер-132),83,132],0)
loadcar(loopер)

```

### 3、字体及字符显示

使用字体模块用来做游戏的文字显示，大部分游戏都会有诸如比分、时间、生命值等的文字信息。pygame 主要是使用 pygame.font 模块来完成，常用到的一些方法是：

pygame.font.SysFont(None, 16)，第一个参数是说明字体的，可以是"arial"等，这里 None 表示默认字体。第二个参数表示字的大小。如果无法知道当前系统中装了哪些字体，可以使用 pygame.font.get\_fonts()来获得所有可用字体。

pygame.font.Font("AAA.ttf", 16)，用来使用 TTF 字体文件。

render("hello world!", True, (0,0,0), (255, 255, 255)),render 方法用来创建文字。第一个参数是写的文字；第二个参数是否开启抗锯齿，就是说 True 的话字体比较平滑，不过相应的速度有一点影响；第三个参数是字体的颜色；第四个是背景色，无表示透明。

下面将上面的例子添加当前汽车坐标：

#@小五义 <http://www.cnblogs.com/xiaowuyi>

```

import pygame,sys
def lineleft():
    plotpoints=[]
    for x in range(0,640):
        y=-5*x+1000        plotpoints.append([x,y])
    pygame.draw.lines(screen,[0,0,0],False,plotpoints,5)
    pygame.display.flip()
def lineright():
    plotpoints=[]
    for x in range(0,640):
        y=5*x-2000        plotpoints.append([x,y])
    pygame.draw.lines(screen,[0,0,0],False,plotpoints,5)
    pygame.display.flip()
def linemiddle():
    plotpoints=[]
    x=300
    for y in range(0,480,20):
        plotpoints.append([x,y])
    if len(plotpoints)==2:
        pygame.draw.lines(screen,[0,0,0],False,plotpoints,5)
        plotpoints=[]
    pygame.display.flip()
def loadcar(yloc):
    my_car=pygame.image.load('ok1.jpg')
    locationxy=[310,yloc]
    screen.blit(my_car,locationxy)
    pygame.display.flip()
def loadtext(xloc,yloc):
    textstr='location:'+str(xloc)+';'+str(yloc)

```

```
text_screen=my_font.render(textstr, True, (255, 0, 0))
screen.blit(text_screen, (50,50))
```

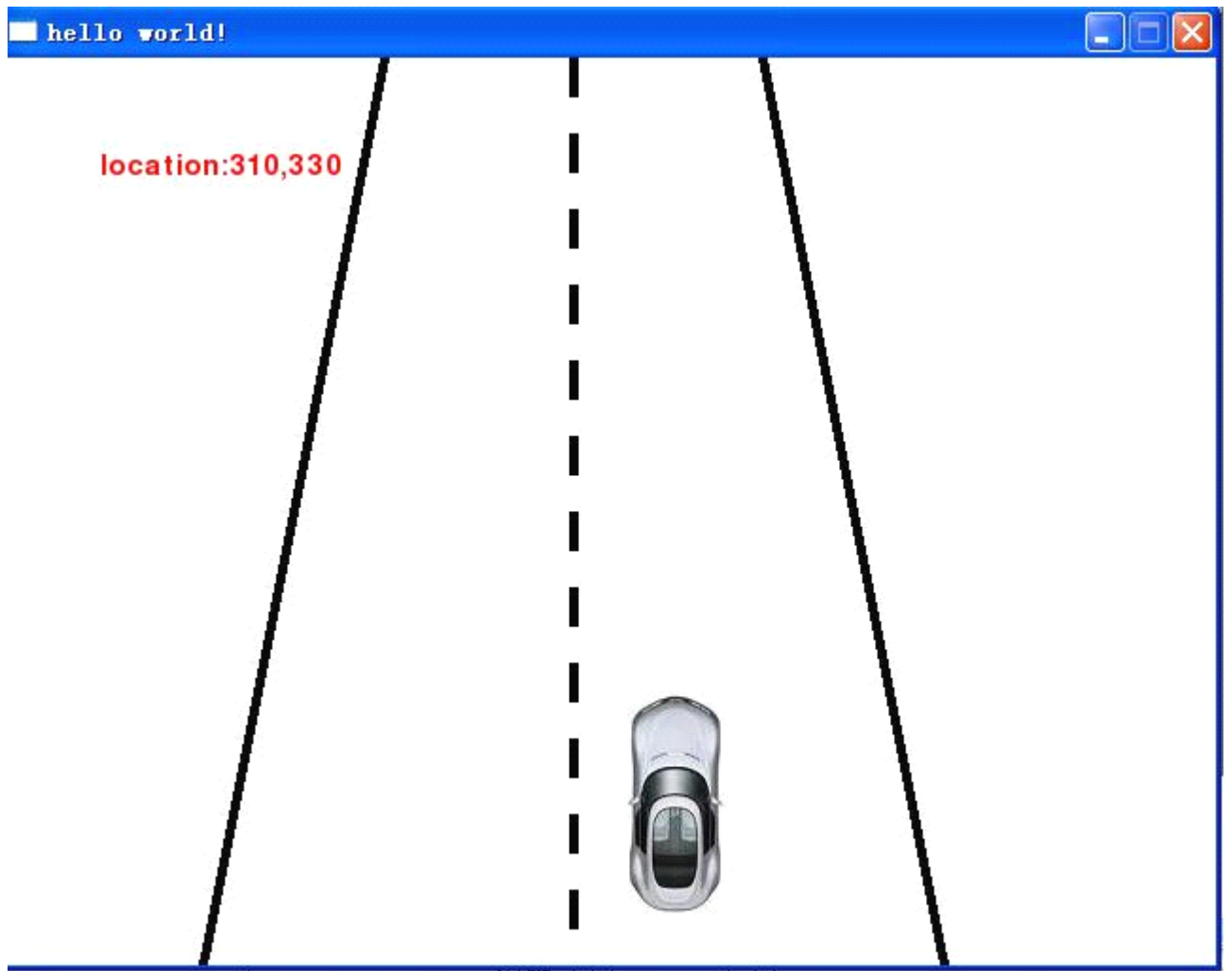
```
if __name__ == '__main__':
    pygame.init()
    screen=pygame.display.set_caption('hello world!')
    screen=pygame.display.set_mode([640,480])

    my_font=pygame.font.SysFont(None,22)
    screen.fill([255,255,255])
    loadtext(310,0)
    lineleft()
    lineright()
    linemiddle()

    looper=480

    while True:
        for event in pygame.event.get():
            if event.type==pygame.QUIT:
                sys.exit()
            elif event.type == pygame.KEYDOWN:
                if event.key==pygame.K_UP:
                    looper=looper-50
                    if looper<=-132:
                        looper=480
                if event.key==pygame.K_DOWN:
                    looper=looper+50
                    if looper>480:
                        looper=-132
                loadtext(310,looper)
            screen.fill([255,255,255])
            loadtext(310,looper)
            lineleft()
            lineright()
            linemiddle()
            loadcar(looper)
```

这个例子里直接让背景重绘一下，就不会再像1、2里面那样用空白的 rect 去覆盖前面的模块了。





## 第四章：声音

pygame.mixer 是一个用来处理声音的模块，其含义为“混音器”。游戏中对声音的处理一般包括制造声音和播放声音两部分，这里仅学习了播放声音部分。

### 1、pygame.mixer 启动与初始化

pygame.mixer.init():mixer 的初始化。程序中，使用时，一般放在最前面的几句代码中：

```
import pygame
pygame.init()
pygame.mixer.init()
```

### 2、播放声音片段 wav 文件

播放声音片段 wav 文件时，pygame.mixer 使用 Sound 对象，格式为：

```
soundwav=pygame.mixer.Sound("filename.wav") #filename.wav 文件名
soundwav.play()
```

在游戏中，往往用下面的代码来完成：

```
#@小五义 http://www.cnblogs.com/xiaowuyi
import pygame,sys
pygame.init()
pygame.mixer.init()
screen=pygame.display.set_mode([640,480])
pygame.time.delay(1000)#等待1秒让 mixer 完成初始化
soundwav=pygame.mixer.Sound("filename.wav")
soundwav.play()
while 1:
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            sys.exit()
```

### 3、播放 mp3、wma、ogg 音乐文件

这里只是学习了一个 mp3 文件，如果非 windows 系统，那可以试试 ogg 文件。播放音乐文件主要使用 pygame.mixer 中的 music 模块，主要方法为：

```
pygame.mixer.music.load("filename.mp3")
pygame.mixer.music.play(n,start,stop)#第一个参数为播放次数，如果是-1表示循环播放，省略表示只播放1次。第二个参数和第三个参数分别表示播放的起始和结束位置。
```

完整的代码为：

```
#@小五义 http://www.cnblogs.com/xiaowuyi
import pygame,sys
pygame.init()
pygame.mixer.init()
screen=pygame.display.set_mode([640,480])
pygame.time.delay(1000)
pygame.mixer.music.load("filename.mp3")
pygame.mixer.music.play()
while 1:
    for event in pygame.event.get():
```

```
if event.type==pygame.QUIT:
    sys.exit()
```

程序在运行时，一旦开始播放音乐，就会继续做下一件事情，也就是说，如果我们要播放两首歌，那么

```
pygame.mixer.music.load("filename1.mp3")
pygame.mixer.music.play()
pygame.mixer.music.load("filename2.mp3")
pygame.mixer.music.play()
```

这段代码运行后，就会出现两首歌同时在播放的情况。实际中，我们常常需要播放完一首，再播放另一首，这样就需要利用 `pygame.mixer.music.get_busy()` 函数来判断，如果函数返回值为 `True` 那么说明目前处于忙的状态，否则返回 `False`。例如：

#@小五义 <http://www.cnblogs.com/xiaowuyi>

```
import pygame,sys
pygame.init()
pygame.mixer.init()
screen=pygame.display.set_mode([640,480])
pygame.time.delay(1000)
pygame.mixer.music.load("filename.mp3")
pygame.mixer.music.play()
soundwav=pygame.mixer.Sound("filename.wav")
while 1:
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            sys.exit()
    if not pygame.mixer.music.get_busy():
        soundwav.play()
        pygame.time.delay(5000)#等待5秒让 filename.wav 播放结束
        sys.exit()
```

#### 4、控制音量

`pygame.mixer.music.set_volume()` 用来控制音量，取值范围为0-1.0的浮点数。0为最小值，1为最大值。

#### 5、制作 mp3播放器中遇到的问题

python+wxpython+pygame 完全可以很容易方便的做出一个 mp3播放器，但是在制作中遇到两个问题：

一是中文歌曲名的显示，通过搜索已经解决，加上 `# encoding='GBK'`

二是 `pygame` 在播放 mp3文件时，发现并不是所有的 mp3文件都有播放，常常会出现播放声音不正确的情况，有时只是杂响。经过与 python 学习交流群 (2)中的 `す easy げ` 交流，`す easy げ` 提出位速128kbps 的 mp3文件可以播放，但从网上下了几个128kbps 的 mp3测试后，发现播放不正常 的情况依然存在。不知道哪位大牛能解决这一问题，希望能把解决方法告知一下，谢谢。

因此，如果您打算写一个 mp3的播放工具，python 能做 mp3播放的模块很多，这里，我觉得可以考虑使用 `mp3play` 模块，比较方便简单。目前版本为 `mp3play 0.1.15`，具体下载地址为 <http://pypi.python.org/pypi/mp3play/>。这里简单介绍一下其主要方法：

`play(n,starms,endms)`:播放，第一个参数表示次数，第二、三个参数表示播放的起止位置，单

位是毫秒。

stop(): 停止。

pause(), unpause(): 暂停和开始。

isplaying(): 判断是否正在播放, 如果返回为 TRUE, 表示正在播放歌曲。

ispaused(): 判断是否为暂停, 如果返回为 TRUE, 表示暂停。

seconds(): 返回 mp3 文件共多少秒, 注意这里的单位是秒。

milliseconds(): 返回 mp3 文件共多少毫秒, 注意这里的单位是毫秒。

mp3play 首页上给出的例子为:

#@小五义 <http://www.cnblogs.com/xiaowuyi>

```
import mp3play
```

```
filename= r'C:\music.mp3'#播放 c 盘下的 music.mp3 文件
```

```
clip = mp3play.load(filename)
```

```
clip.play()
```

```
import time
```

```
time.sleep(min(30, clip.seconds()))#如果 mp3 文件的长度小于30秒时, 全部播放完, 否则仅播放30秒。
```

```
clip.stop()
```

## 第五章：精灵

据说在任天堂 FC 时代，精灵的作用相当巨大，可是那时候只知道怎么玩超级玛丽、魂斗罗，却对精灵一点也不知。pygame.sprite.Sprite 就是 Pygame 里面用来实现精灵的一个类，使用时，并不需要对它实例化，只需要继承他，然后按需写出自己的类 就好了，因此非常简单实用。

### 一、什么是精灵

精灵可以认为成是一个个小图片，一种可以在屏幕上移动的图形对象，并且可以与其他图形对象交互。精灵图像可以是使用 pygame 绘制函数绘制的图像，也可以是原来就有的图像文件。

### 二、sprite 中主要且常用的变量

(更多详细的见 <http://www.pygame.org/docs/ref/sprite.html#pygame.sprite.Sprite>)

self.image 这个负责显示什么。如 self.image=pygame.Surface([x,y])说明该精灵是一个 x,y 大小的距 形，self.image=pygame.image.load(filename)说明该精灵调用显示 filename 这个图片文件。

self.image.fill([color]), 负责对 self.image 着色，如 self.image=pygame.Surface([x,y])

self.image.fill([255,0,0])

对 x,y 距形填充红色。

self.rect 负责在哪里显示。一般来说，先用 self.rect=self.image.get\_rect()获得 image 距形大小，然后给 self.rect 设定 显示的位置，一般用 self.rect.topleft (topright、bottomleft、bottomright) 来设定某一个角的显示位置。另外，self.rect.top、self.rect.bottom、self.rect.right、self.rect.left 分别表示上下左 右。

self.update 负责使精灵行为生效。

Sprite.add 添加精灵到 group 中去。

Sprite.remove 从 group 中删除

Sprite.kill 从 groups 中全部删除精灵

Sprite.alive 判断精灵是否属于 groups

### 三、建立一个简单的精灵

所有精灵在建立时都是从 pygame.sprite.Sprite 中继承的。

(1) 做一个精灵，绘制一个宽30、高30的距形，具体代码如下：

```
class Temp(pygame.sprite.Sprite):
    def __init__(self,color,initial_position):
        pygame.sprite.Sprite.__init__(self)
        self.image= pygame.Surface([30,30])
        self.image.fill(color)
        self.rect=self.image.get_rect()
        self.rect.topleft=initial_position
```

这里逐句进行一下分析，pygame.sprite.Sprite.\_\_init\_\_(self)完成初始化。self.image = pygame.Surface([30,30])定义显示30\*30的一个距形 surface。self.image.fill(color)用 color 来填充颜色。self.rect=self.image.get\_rect()获取 self.image 大小。 self.rect.topleft=initial\_position 确定左上角显示位置，当然也可以用 topright、bottomrigh、 bottomleft 来分别确定其他几个角的位置。精灵的显示，在一个640\*480大小的白色窗体[50,100]的位置绘制一个30\*30大小

的红色 矩形，完整代码如下：

#小五义 <http://www.cnblogs.com/xiaowuyi>

```
import pygame,sys
pygame.init()
class Temp(pygame.sprite.Sprite):
    def __init__(self,color,initial_position):
        pygame.sprite.Sprite.__init__(self)
        self.image= pygame.Surface([30,30])
        self.image.fill(color)
        self.rect=self.image.get_rect()
        self.rect.topleft=initial_position
screen=pygame.display.set_mode([640,480])
screen.fill([255,255,255])
b=Temp([255,0,0],[50,100])
screen.blit(b.image,b.rect)
pygame.display.update()
while True:
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            sys.exit()
```

(2) 做一个精灵，显示内容为某一图片，这里以前面用过的小车图片为例，代码如下：

#小五义 <http://www.cnblogs.com/xiaowuyi>

```
import pygame,sys
pygame.init()
class Car(pygame.sprite.Sprite):
    def __init__(self,filename,initial_position):
        pygame.sprite.Sprite.__init__(self)
        self.image=pygame.image.load(filename)
        self.rect=self.image.get_rect()
        #self.rect.topleft=initial_position
        self.rect.bottomright=initial_position
        print self.rect.right
```

```
screen=pygame.display.set_mode([640,480])
screen.fill([255,255,255])
fi='ok1.jpg'b=Car(fi,[150,100])
screen.blit(b.image,b.rect)
pygame.display.update()
while True:
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            sys.exit()
```

这段代码与（1）的不同之处在于 self.image 定义为 pygame.image.load(filename)，用来显示 filename 文件，本代码使用了 ok1.jpg 文件，并定义了小车右底角的显示位置是[150,100]。

### 三、学习精灵组

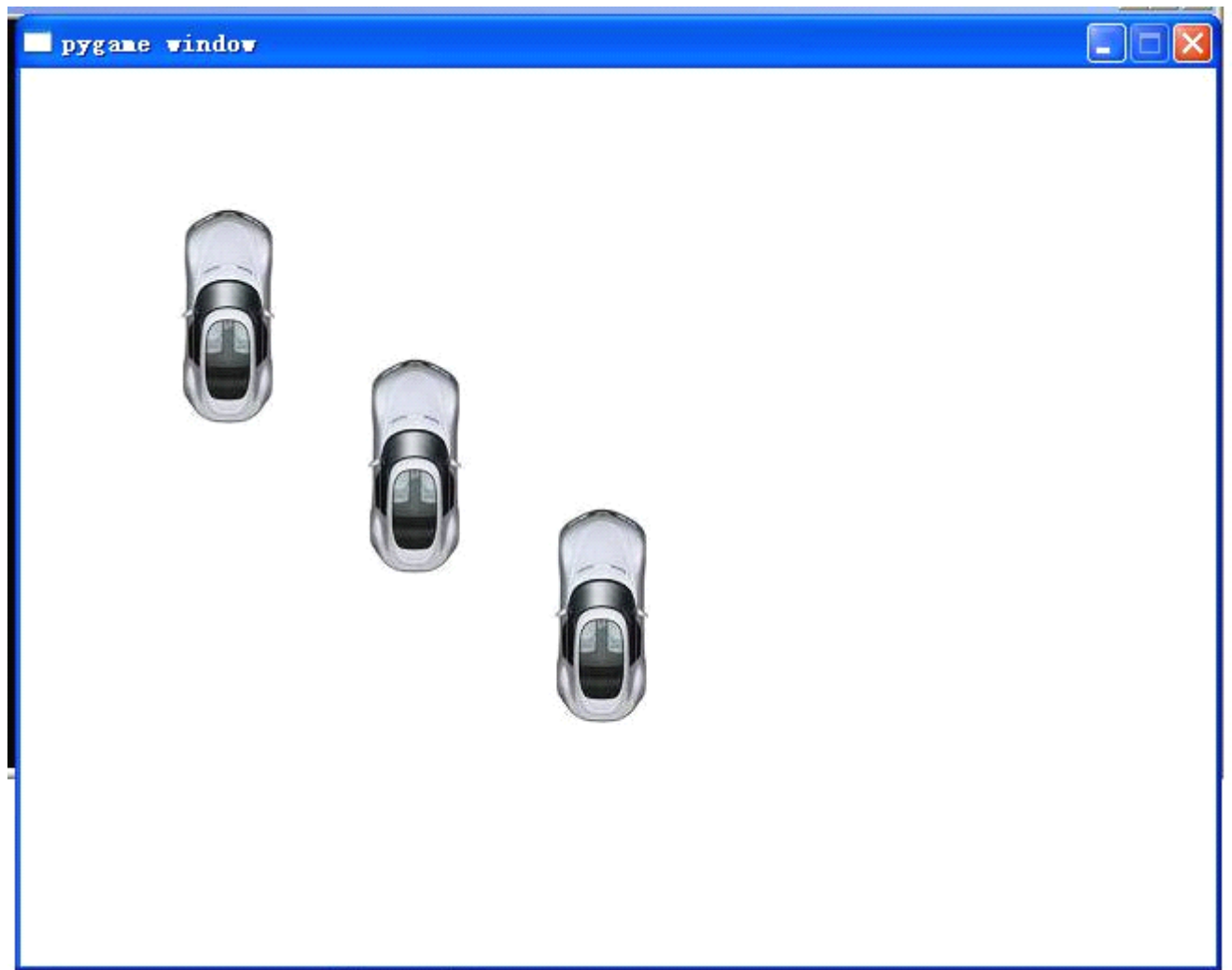
(1) 使用精灵在屏幕上放多个图像，这种方法没用利用精灵组的概念，而是利用了 list 来生成每一个精灵。Cargroup 用来存储不同位置的 Car，screen.blit(carlist.image,carlist.rect) 逐个显示每一个精灵。具体见代码：

#小五义 <http://www.cnblogs.com/xiaowuyi>

```
import pygame,sys
pygame.init()
class Car(pygame.sprite.Sprite):
    def __init__(self,filename,initial_position):
        pygame.sprite.Sprite.__init__(self)
        self.image=pygame.image.load(filename)
        self.rect=self.image.get_rect()
        self.rect.bottomright=initial_position

screen=pygame.display.set_mode([640,480])
screen.fill([255,255,255])
fi='ok1.jpg'locationgroup=([150,200],[350,360],[250,280])
Cargroup=[]
for lo in locationgroup:
    Cargroup.append(Car(fi,lo))
for carlist in Cargroup:
    screen.blit(carlist.image,carlist.rect)
pygame.display.update()
while True:
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            sys.exit()
```

具体效果见图：



(2)使用精灵组来实现多个图像。上面精灵是存在一个列表中，很方便，就是有点不太好用。除了精灵，pygame 还提供了精灵组，它很适合处理精灵列表，有添加，移除，绘制，更新等方法。具体如下：<http://www.pygame.org/docs/ref/sprite.html#pygame.sprite.Sprite>

Group.sprites 精灵组

Group.copy 复制

Group.add 添加

Group.remove 移除

Group.has 判断精灵组成员

Group.update 更新

Group.draw 位块显示

Group.clear - 绘制背景

Group.empty 清空

同样还是上面的这个例子，这里用精灵组来实现。

#小五义 <http://www.cnblogs.com/xiaowuyi>

```
import pygame,sys
```

```
pygame.init()
```

```
class Car(pygame.sprite.Sprite):
```

```
    def __init__(self,filename,initial_position):
```

```

pygame.sprite.Sprite.__init__(self)
self.image=pygame.image.load(filename)
self.rect=self.image.get_rect()
self.rect.bottomright=initial_position
screen=pygame.display.set_mode([640,480])
screen.fill([255,255,255])
fi='ok1.jpg'locationgroup=([150,200],[350,360],[250,280])
Cargroup=pygame.sprite.Group()
for lo in locationgroup:
    Cargroup.add(Car(fi,lo))

```

```

for carlist in Cargroup.sprites():
    screen.blit(carlist.image,carlist.rect)
pygame.display.update()
while True:
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            sys.exit()

```

两个例子都是在[150,200],[350,360],[250,280]三个位置显示三辆小车，不同之处第一个用的是 list，第二个用的是精灵组。差别就在几句话上，一是 Cargroup=pygame.sprite.Group() 定义 Cargroup 为精灵组，二是 Cargroup.add(Car(fi,lo)) 用 add 代替了 append，三是 for carlist in Cargroup.sprites() 这句中逐个显示精灵，这里试了一下，直接用 for carlist in Cargroup 也是可以的。精灵组的代码是高度优化过了，常常比列表还快。插入和删除都是常见的操作，代码还可以避免内存存在循环中反复消耗。

#### 四、动画

利用精灵组做动画会显得比较方便，这里我们首先让上面的三辆小车运动起来。

(1) 三辆小车以不同的速度前行，利用 random.choice 随机生成[-10,-1]之间的值作为速度让小车从下向上运动，并且当到达顶部时，再从底部出现。代码如下：

#小五义 <http://www.cnblogs.com/xiaowuyi>

```

import pygame,sys
from random import *pygame.init()
class Car(pygame.sprite.Sprite):
    def __init__(self,filename,initial_position,speed):
        pygame.sprite.Sprite.__init__(self)
        self.image=pygame.image.load(filename)
        self.rect=self.image.get_rect()
        self.rect.topleft=initial_position
        self.speed=speed
    def move(self):
        self.rect=self.rect.move(self.speed)
        if self.rect.bottom < 0:    #当小车底部到达窗口顶部时，让小车从下面出来
            self.rect.top=480screen=pygame.display.set_mode([640,480])
screen.fill([255,255,255])
fi='ok1.jpg'locationgroup=([150,200],[350,300],[250,200])

```



```

Cargroup=pygame.sprite.Group()
for lo in locationgroup:
    speed=[0,choice([-10,-1])]
    Cargroup.add(Car(fi,lo,speed))

```

while True:

```

    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            sys.exit()
    pygame.time.delay(20)
    screen.fill([255,255,255])
    for carlist in Cargroup.sprites():
        carlist.move()
        screen.blit(carlist.image,carlist.rect)
    pygame.display.update()

```

(2) 可以通过左右键控制三辆小车的左右移动，按左键向左移动，当到达最左边时，不再移动，按右键向右移动，当到达最右边时，不再移动。具体代码如下：

#小五义 <http://www.cnblogs.com/xiaowuyi>

```

import pygame,sys
from random import *
pygame.init()
class Car(pygame.sprite.Sprite):
    def __init__(self,filename,initial_position,speed):
        pygame.sprite.Sprite.__init__(self)
        self.image=pygame.image.load(filename)
        self.rect=self.image.get_rect()
        self.rect.topleft=initial_position
        self.speed=speed
    def move(self):
        self.rect=self.rect.move(self.speed)
        if self.rect.bottom < 0:
            self.rect.top=480
    def moveleft(self):
        self.rect.left=self.rect.left-10
        if self.rect.left<0:
            self.rect.left=0
    def moveright(self):
        self.rect.right=self.rect.right+10
        if self.rect.right>640:
            self.rect.right=640
screen=pygame.display.set_mode([640,480])
screen.fill([255,255,255])
fi='ok1.jpg'locationgroup=([150,200],[350,300],[250,200])
Cargroup=pygame.sprite.Group()
for lo in locationgroup:
    speed=[0,choice([-10,-1])]

```

```
Cargroup.add(Car(fi,lo,speed))
```

```
while True:
```

```
    for event in pygame.event.get():
```

```
        if event.type==pygame.QUIT:
```

```
            sys.exit()
```

```
        elif event.type == pygame.KEYDOWN:
```

```
            if event.key==pygame.K_LEFT:
```

```
                for carlist in Cargroup.sprites():
```

```
                    carlist.moveleft()
```

```
                    screen.blit(carlist.image,carlist.rect)
```

```
            if event.key==pygame.K_RIGHT:
```

```
                for carlist in Cargroup.sprites():
```

```
                    carlist.moveright()
```

```
                    screen.blit(carlist.image,carlist.rect)
```

```
pygame.time.delay(20)
```

```
screen.fill([255,255,255])
```

```
for carlist in Cargroup.sprites():
```

```
    carlist.move()
```

```
    screen.blit(carlist.image,carlist.rect)
```

```
pygame.display.update()
```

## 第六章：一个超级简单的游戏

学了这么长时间的 Pygame，一直想写个游戏实战一下。看起来很简单游戏，写起来怎么这么难。最初想写个俄罗斯方块，想了很长时间如何实现，想来想去，也没写出来，于是干脆下载别人的代码来读。后来，要想写一个帮助记忆的挖宝箱的游戏，结果也没完成。唯一完成了就是下面这个小人接金币的游戏，超级简单，通过左右键控制小人移动去接空中下下来的金币，接住金币得5分，接不住游戏结束，金币速度会随着 level 的关数而越来越快。完成这段代码后，我依然觉得这段代码写得很差，确实也是自己对 pygame 只是掌握了皮毛，对 surface、sprite 这些理解的还不透彻。这里把代码写出来，有时间的大牛们可以帮助指点一下，让我也有所提高。

```
# -*- coding: cp936 -*-
```

```
"""@小五义 http://www.cnblogs.com/xiaowuyi
```

```
一个超级简单的游戏
```

```
左右键控制小人移动去接空中下下来的金币，接住金币得5分，接不住游戏结束，金币速度会  
随着 level 的关数  
而越来越快
```

```
"""
```

```
import pygame,sys,os,random
```

```
pygame.init()
```

```
class rect():#画出小人
```

```
    def __init__(self,filename,initial_position):  
        self.image=pygame.image.load(filename)  
        self.rect=self.image.get_rect()  
        self.rect.topleft=initial_position
```

```
class goldrect(pygame.sprite.Sprite):#绘出金币
```

```
    def __init__(self,gold_position,speed):  
        pygame.sprite.Sprite.__init__(self)  
        self.image=pygame.image.load('image\\gold.png')  
        self.rect=self.image.get_rect()  
        self.rect.topleft=gold_position  
        self.speed=speed
```

```
    def move(self):  
        self.rect=self.rect.move(self.speed)
```

```
def drawback(): #绘出背景图片
```

```
    my_back=pygame.image.load('image\\qi3.jpg')  
    bakscreen.blit(my_back,[0,0])
```

```
def loadtext(levelnum,score,highscore):#绘出成绩、level、最高分等
```

```
    my_font=pygame.font.SysFont(None,24)  
    levelstr='Level:'+str(levelnum)  
    text_screen=my_font.render(levelstr, True, (255, 0, 0))  
    bakscreen.blit(text_screen, (650,50))  
    highscorestr='Higescore:'+str(highscore)
```

```

text_screen=my_font.render(highscorestr, True, (255, 0, 0))
bakscreen.blit(text_screen, (650,80))
scorestr='Score:'+str(score)
text_screen=my_font.render(scorestr, True, (255, 0, 0))
bakscreen.blit(text_screen, (650,110))
def loadgameover(scorenum,highscore):#绘出 GAME OVER
    my_font=pygame.font.SysFont(None,50)
    levelstr='GAME OVER'    over_screen=my_font.render(levelstr, True, (255, 0, 0))
    bakscreen.blit(over_screen, (300,240))
    highscorestr='YOUR SCORE IS'+str(scorenum)
    over_screen=my_font.render(highscorestr, True, (255, 0, 0))
    bakscreen.blit(over_screen, (280,290))
    if scorenum>int(highscore):#写入最高分
        highscorestr='YOUR HAVE GOT THE HIGHEST SCORE!'
    text_screen=my_font.render(highscorestr, True, (255, 0, 0))
    bakscreen.blit(text_screen, (100,340))
    highfile=open('highscore','w')
    highfile.writelines(str(scorenum))
    highfile.close()
def gethighscore(): #读取最高分
    if os.path.isfile('highscore'):
        highfile=open('highscore','r')
        highscore=highfile.readline()
        highfile.close()
    else:
        highscore=0
    return highscore
bakscreen=pygame.display.set_mode([800,600])
bakscreen.fill([0,160,233])
pygame.display.set_caption('Dig!Dig!')
drawback()

```

```

levelnum=1 #level
scorenum=0 #得分
highscore=gethighscore()#最高分
ileft=1 #记录向左移动步数，用来控制图片
iright=10 #记录向右移动步数，用来控制图片
x=100y=450filename='image\\1.png'backimg_ren=rect(filename,[x,y])
bakscreen.blit(backimg_ren.image,backimg_ren.rect)
loadtext(levelnum,scorenum,highscore)
goldx=random.randint(50,580)
speed=[0,levelnum]

```

```
mygold=goldrect([goldx,100],speed)
pygame.display.update()
```

```
while True:
```

```
    if scorenum>0 and scorenum/50.0==int(scorenum/50.0):#当得分是50的倍数时修改 level
        levelnum=scorenum/50+1        speed=[0,levelnum]
```

```
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            sys.exit()
```

```
    #make gold
```

```
    pressed_keys= pygame.key.get_pressed()
    if pressed_keys[pygame.K_LEFT]:#按下左键
        drawback()
        loadtext(levelnum,scorenum,highscore)
    if iright > 14 :iright=10
    iright=iright+1
    filename='image\\'+str(iright)+'.png'
    if x<50 :
        x=50
    else:
        x=x-10
    backing_surface=rect(filename,[x,y])
    bakscreen.blit(backimg_surface.image,backimg_surface.rect)
    if pressed_keys[pygame.K_RIGHT]:#按下右键
        drawback()
        loadtext(levelnum,scorenum,highscore)
    if ileft > 4 :ileft=0
    ileft=ileft+1        filename='image\\'+str(ileft)+'.png'
    if x>560:
        x=560
    else:
        x=x+10
    backing_surface=rect(filename,[x,y])
    bakscreen.blit(backimg_surface.image,backimg_surface.rect)
    drawback()
    loadtext(levelnum,scorenum,highscore)
    mygold.move()
    bakscreen.blit(mygold.image,mygold.rect)

    backing_surface=rect(filename,[x,y])
    bakscreen.blit(backimg_surface.image,backimg_surface.rect)
    if mygold.rect.top>600:#判断金币是否着地，一但着地，游戏结束
        loadgameover(scorenum,highscore)
```

```
if mygold.rect.colliderect(backimg_surface.rect):#判断金币是否与小人碰撞，如果碰撞表示  
小人接到金币
```

```
    scorenum+=5        loadtext(levelnum,scorenum,highscore)  
    goldx=random.randint(50,580)  
    mygold=goldrect([goldx,100],speed)  
pygame.display.update()
```

