

浦东新区2023年Python教师培训项目

第6课-Python的常用库

课程概要

pygame是被设计用来写游戏的python模块集合，pygame是在优秀的SDL库之上开发的功能性包。使用python可以导入pygame来开发具有全部特性的游戏和多媒体软件，pygame是极度轻便的并且可以运行在几乎所有的平台和操作系统上。

图形用户界面（GraphicalUserInterface，简称GUI，又称图形用户接口）是指采用图形方式显示的计算机操作用户界面。图形用户界面是一种人与计算机通信的界面显示格式，允许用户使用鼠标等输入设备操纵屏幕上的图标或菜单选项，以选择命令、调用文件、启动程序或执行其它一些日常任务。与通过键盘输入文本或字符命令来完成例行任务的字符界面相比，图形用户界面有许多优点。图形用户界面由窗口、下拉菜单、对话框及其相应的控制机制构成，在各种新式应用程序中都是标准化的，即相同的操作总是以同样的方式来完成，在图形用户界面，用户看到和操作的都是图形对象，应用的是计算机图形学的技术。本节课我们将使用tkinter来实现图形用户界面。

知识点

如何安装第三方库

通过命令行方式安装

点击键盘win+R键，输入cmd打开命令提示符，输入pip install pygame即可安装。

pygame常用函数及方法

```
# -----窗口相关操作-----
# 创建窗口
scene = pygame.display.set_mode([屏幕宽, 屏幕高])
# 设置窗口标题
pygame.display.set_caption("标题")
# 设置窗口图标
pygame.display.set_icon(image)
# 将图片绘制到窗口
scene.blit(image, (0, 0))

# -----图像相关操作-----
# 加载图片
```

```

image = pygame.image.load("图片路径")
# 获得图片矩形 -> Rect(x, y, width, height)
rect = image.get_rect()
# 移动矩形坐标
rect.move_ip(x, y)
# 判断两个矩形是否相交 -> bool
flag = pygame.Rect.colliderect(rect1, rect2)

# -----事件相关操作-----
# 获得所有事件列表
event_list = pygame.event.get()
# 常见事件类型
# QUIT 关闭窗口
# MOUSEMOTION 鼠标拖动
# KEYDOWN 键盘按键
# 获得当前所有持续按键 bools_tuple
bool_list = pygame.key.get_pressed()

# -----音效相关操作-----
# 加载背景音乐
pygame.mixer.music.load("./res/bg2.ogg")
# 播放背景音乐, -1表示循环播放
pygame.mixer.music.play(-1)
# 加载音效
self.bomb = pygame.mixer.Sound("./res/baozha.ogg")
# 播放音效
pygame.mixer.Sound.play(self.bomb)

```

课堂案例

雪花下落

本案例是使用pygame模块来生成窗口、背景、雪花，实现雪花不断下落的效果。

```

import pygame #导入工具箱
import random #随机数工具箱
pygame.init() #初始化工具箱

x=[] #存放100个雪花的x坐标
y=[] #存放100个雪花的y坐标
snow=[] #存放100个雪花

#第一步：生成一个窗口
sc=pygame.display.set_mode((700,460)) #生成一个标准窗口 screen

#第二步：贴背景
bg=pygame.image.load("snow.png") #2.1 加载图片 background

while True: #一直重复做某件事情
    sc.blit(bg,(0,0)) #2.2 贴图片 把...贴到。。。上

```

```

#第三步：贴雪花
for i in range(100):
    ft=pygame.font.Font("simsun.ttc",random.randint(10,60)) #3.1 创建一种字体 font
    sw=ft.render("*",True,(255,255,255)) #3.2 生成文字 snow
    snow.append(sw)

for i in range(100): #重复100次
    x.append(random.randint(0,700)) #添加对应的x坐标
    y.append(random.randint(0,460)) #添加对应的y坐标
    sc.blit(snow[i],(x[i],y[i])) #3.3 贴雪花

for i in range(100):
    y[i]=y[i]+2 #雪花下落, y坐标增加
    if y[i]>460: #如果雪花落到最下面, 就把y坐标变成0
        y[i]=0
        x.append(random.randint(0,700))

#刷新
pygame.display.update() #刷新

```

猴子接水果

本案例是使用pygame模块来生成窗口、贴背景、猴子移动、水果下落以及键盘侦测效果的案例。

```

import pygame,time,random
from pygame.locals import * #从pygame本地导入所有函数或者对象
from sys import exit #sys-->system 从系统中导入退出方法
pygame.init()
pygame.key.set_repeat(1,1) #重复获取鼠标事件

sc = pygame.display.set_mode((900,650)) #创建一个窗口
bg = pygame.image.load("background.png") #背景=加载一个背景图片
bg = pygame.transform.scale(bg,(900,650)) #背景=缩放一个背景图片

pygame.display.set_caption('猴子接水果') #在窗口显示游戏名称'猴子接水果'

monkey = pygame.image.load("monkey-a.png") #加载猴子图片
monkey = pygame.transform.scale(monkey,(80,123)) #缩放猴子图片
monkeyx=400 #创建猴子的x坐标
monkeyy=500 #创建猴子的y坐标

monkey_b1 = pygame.image.load("monkey-b1.png") #加载猴子右转图片
monkey_b1 = pygame.transform.scale(monkey_b1,(80,123)) #缩放猴子右转图片
monkey_b2 = pygame.image.load("monkey-b2.png") #加载猴子左转图片
monkey_b2 = pygame.transform.scale(monkey_b2,(80,123)) #缩放猴子左转图片

lemon = pygame.image.load("lemon.png") #加载柠檬图片
lemon = pygame.transform.scale(lemon,(42,40)) #缩放柠檬图片
lemon_x=[] #创建柠檬的x坐标列表
lemon_y=[] #创建柠檬的y坐标列表

apple = pygame.image.load("apple.png") #加载苹果图片

```

```

apple = pygame.transform.scale(apple,(42,40)) #缩放苹果图片
apple_x=[] #创建苹果的x坐标列表
apple_y=[] #创建苹果的y坐标列表

wm = pygame.image.load("watermelon.png") #加载西瓜图片
wm = pygame.transform.scale(wm,(50,40)) #缩放西瓜图片
wm_x=[] #创建西瓜的x坐标列表
wm_y=[] #创建西瓜的y坐标列表

cherry = pygame.image.load("cherry.png") #加载西瓜图片
cherry = pygame.transform.scale(cherry,(50,50)) #缩放西瓜图片
cherry_x=[] #创建樱桃的x坐标列表
cherry_y=[] #创建樱桃的y坐标列表

isgameover=0 #是否游戏结束的标记, 当isgameover为0时, 游戏不结束; 当isgameover为1时, 游戏结束
gameover=pygame.image.load('gameover.png') #加载游戏结束图片
gameover=pygame.transform.scale(gameover,(900,650))

font=pygame.font.Font('simsun.ttc',40) #创建一种字体
text=font.render('得分:',True,(255,0,0)) #创建文本
score=0 #分数初始化为0

movex=0 #变化的x坐标

times=3000 #游戏时间

#-----主循环体, 这是程序真正实现效果的地方-----
while True:
    sc.blit(bg,(0,0))
    sc.blit(text,(10,10)) #将文本贴到窗口中
    text_score=font.render(str(score),True,(255,0,0))
    sc.blit(text_score,(120,10)) #将分数贴到窗口中
    #游戏正常运行
    if isgameover==0:
        monkeyX+=movex
        if movex<0:
            sc.blit(monkey_b2,(monkeyX,monkeyY))
            #pygame.display.update()
        elif movex>0:
            sc.blit(monkey_b1,(monkeyX,monkeyY))
            #pygame.display.update()
        else:
            sc.blit(monkey,(monkeyX,monkeyY))
        times-=1
        if times==0:
            isgameover=1
    #-----添加水果-----
    #times每减少60就往柠檬坐标列表里添加一个坐标
    if times%60 == 0:
        lemon_x.append(random.randint(0,900-lemon.get_width()))
        lemon_y.append(0)

    #times每减少80就往苹果坐标列表里添加一个坐标

```

```

if times%80 == 0:
    apple_x.append(random.randint(0,900-apple.get_width()))
    apple_y.append(0)

#times每减少100就往西瓜坐标列表里添加一个坐标
if times%100 == 0:
    wm_x.append(random.randint(0,900-wm.get_width()))
    wm_y.append(0)

#times每减少120就往西瓜坐标列表里添加一个坐标
if times%100 == 0:
    cherry_x.append(random.randint(0,900-cherry.get_width()))
    cherry_y.append(0)

#-----贴水果-----
#贴所有柠檬的坐标
for i in range(len(lemon_x)):
    sc.blit(lemon,(lemon_x[i],lemon_y[i]))

#贴所有苹果的坐标
for i in range(len(apple_x)):
    sc.blit(apple,(apple_x[i],apple_y[i]))

#贴所有西瓜的坐标
for i in range(len(wm_x)):
    sc.blit(wm,(wm_x[i],wm_y[i]))

#贴所有樱桃的坐标
for i in range(len(cherry_x)):
    sc.blit(cherry,(cherry_x[i],cherry_y[i]))

#-----水果下落-----
#实现柠檬下落的方法
for i in range(len(lemon_x)):
    lemon_y[i]=lemon_y[i]+2
    if lemon_y[i]>650:
        del lemon_x[i] #del==>delete删除
        del lemon_y[i]
        break

#实现苹果下落的方法
for i in range(len(apple_x)):
    apple_y[i]=apple_y[i]+2
    if apple_y[i]>650:
        del apple_x[i] #del==>delete删除
        del apple_y[i]
        break

#实现西瓜下落的方法
for i in range(len(wm_x)):
    wm_y[i]=wm_y[i]+2
    if wm_y[i]>650:
        del wm_x[i] #del==>delete删除

```

```

        del wm_y[i]
        break

#实现樱桃下落的方法
for i in range(len(cherry_x)):
    cherry_y[i]=cherry_y[i]+2
    if cherry_y[i]>650:
        del cherry_x[i] #del==>delete删除
        del cherry_y[i]
        break

#-----水果被吃掉-----
#实现柠檬被吃掉的方法
for i in range(len(lemon_x)):
    if lemon_x[i]>monkeyX and lemon_x[i]<monkeyX+monkey.get_width()-
lemon.get_width():
        if lemon_y[i]>monkeyY-lemon.get_height() and lemon_y[i]
<monkeyY+monkey.get_height():
            score = score+1
            del lemon_x[i]
            del lemon_y[i]
            break
#实现苹果被吃掉的方法
for i in range(len(apple_x)):
    if apple_x[i]>monkeyX and apple_x[i]<monkeyX+monkey.get_width()-
apple.get_width():
        if apple_y[i]>monkeyY-apple.get_height() and apple_y[i]
<monkeyY+monkey.get_height():
            score = score+1

            del apple_x[i]
            del apple_y[i]
            break
#实现西瓜被吃掉的方法
for i in range(len(wm_x)):
    if wm_x[i]>monkeyX and wm_x[i]<monkeyX+monkey.get_width()-wm.get_width():
        if wm_y[i]>monkeyY-wm.get_height() and wm_y[i]<monkeyY+monkey.get_height():
            score = score+1
            del wm_x[i]
            del wm_y[i]
            break

#实现樱桃被吃掉的方法
for i in range(len(cherry_x)):
    if cherry_x[i]>monkeyX and cherry_x[i]<monkeyX+monkey.get_width()-
wm.get_width():
        if cherry_y[i]>monkeyY-cherry.get_height() and cherry_y[i]
<monkeyY+monkey.get_height():
            score = score+1
            del cherry_x[i]
            del cherry_y[i]
            break

```

```

elif isgameover==1:
    sc.blit(gameover,(0,0))

else:
    time=3000
    lemon_x=[]    #创建柠檬的X坐标列表
    lemon_y=[]    #创建柠檬的Y坐标列表
    apple_x=[]    #创建苹果的X坐标列表
    apple_y=[]    #创建苹果的Y坐标列表
    wm_x=[]       #创建西瓜的X坐标列表
    wm_y=[]       #创建西瓜的Y坐标列表
    cherry_x=[]   #创建樱桃的X坐标列表
    cherry_y=[]   #创建樱桃的Y坐标列表
    score=0       #分数清零
    monkeyX=400   #创建猴子的X坐标
    monkeyY=500   #创建猴子的Y坐标
    isgameover=0

pygame.display.update()
#实现窗口退出
for event in pygame.event.get():
    if event.type==pygame.QUIT:
        #双等于==代表判断，单个等于=代表赋值
        pygame.quit()
        exit()
    if event.type==KEYDOWN: #键盘按下事件
        if event.key==K_LEFT: #键盘按钮是否为左边
            movex=-10
        if event.key==K_RIGHT: #键盘按钮是否为右边
            movex=10
        if event.key==K_SPACE: #键盘按钮是否为空格
            if isgameover==1:
                isgameover=2 #0代表游戏重新开始
    elif event.type==KEYUP:
        movex = 0

```

什么是tkinter

Tkinter 是 Python 的标准 GUI 库。Python 使用 Tkinter 可以快速的创建 GUI 应用程序。由于 Tkinter 是内置到 python 的安装包中、只要安装好 Python 之后就能 import Tkinter 库、而且 IDLE 也是用 Tkinter 编写而成、对于简单的图形界面 Tkinter 还是能应付自如。

tkinter组件

控件	描述
Button	按钮控件；在程序中显示按钮。
Canvas	画布控件；显示图形元素如线条或文本
Checkbutton	多选框控件；用于在程序中提供多项选择框
Entry	输入控件；用于显示简单的文本内容
Frame	框架控件；在屏幕上显示一个矩形区域，多用来作为容器
Label	标签控件；可以显示文本和位图
Listbox	列表框控件；在Listbox窗口小部件是用来显示一个字符串列表给用户
Menubutton	菜单按钮控件，用于显示菜单项。
Menu	菜单控件；显示菜单栏,下拉菜单和弹出菜单
Message	消息控件；用来显示多行文本，与label比较类似
Radiobutton	单选按钮控件；显示一个单选的按钮状态
Scale	范围控件；显示一个数值刻度，为输出限定范围的数字区间
Scrollbar	滚动条控件，当内容超过可视化区域时使用，如列表框。。
Text	文本控件；用于显示多行文本
Toplevel	容器控件；用来提供一个单独的对话框，和Frame比较类似
Spinbox	输入控件；与Entry类似，但是可以指定输入范围值
PanedWindow	PanedWindow是一个窗口布局管理的插件，可以包含一个或者多个子控件。
LabelFrame	labelframe 是一个简单的容器控件。常用与复杂的窗口布局。
tkMessageBox	用于显示你应用程序的消息框。

标准属性

标准属性也就是所有控件的共同属性，如大小，字体和颜色等等。

属性	描述
Dimension	控件大小；
Color	控件颜色；
Font	控件字体；
Anchor	锚点；
Relief	控件样式；
Bitmap	位图；
Cursor	光标；

几何管理

Tkinter控件有特定的几何状态管理方法，管理整个控件区域组织，以下是Tkinter公开的几何管理类：包、网格、位置

几何方法	描述
pack()	包装
grid()	网格
place()	位置

相关学习网站

Pygame官方文档的链接：<http://www.pygame.org/docs/>

PythonGUI编程(Tkinter): <https://www.runoob.com/python/python-gui-tkinter.html>