**小游戏Ccharles实验报告（Python）**

**一、游戏规则与思路**

**规则：**Ccharles是一款鼠标控制的躲避类小游戏，每隔一定时间增加一个球，级别上升一级，计分以时间为基础，每坚持一秒即可得到一分，另外设置一个星星球，碰到星星球即可得到五分，但是碰到其他颜色（黄色、蓝色、绿色、橙色）的球，则游戏结束，点击鼠标即可开始下一轮的游戏，分数、时间、级别从零开始计。

**思路：**想制作这个游戏是因为之前玩儿过一个手机游戏Charles，所以仿照这个游戏用Python（pygame）编写了Ccharles，在界面、音乐和画风上都仿照了Charles，但是规则略有不同，因为水平有限，并没有设置很多道具，只完成了最基础的躲避规则。游戏的实现过程如下：一旦游戏窗口打开，则开始播放背景音乐（foolish game），计时器启动，每隔一定时间生成一个新球，初始为第一个黄球和一个星星球，之后按照绿、橙、蓝、黄的顺序依次循环生成新球，这些颜色的球不会消失，会在界面按初始方向运动，若碰到边界则反弹继续运动，而星星球只在被鼠标碰到后立即重新生成，运动方式和其他球一致，每次碰到星星球，则计分器加五分，计时器每过一秒计分器加一分。（【注】在游戏结束后继续进行游戏，球产生的初始位置会接着上一局的移动，且新生成球为上一局最后一个球的下一个，因为游戏是在while（1）的循环内进行的。）

**二、游戏主逻辑**

######游戏逻辑######

while 1:

#clock.tick(100)

screen.blit(background, (0,0))

#获得鼠标的xy坐标

xy = pygame.mouse.get\_pos()

#mousex=pygame.mouse.get\_pos()

#mousey=pygame.mouse.get\_pos()

#隐藏鼠标

pygame.mouse.set\_visible(False)

#获取光标图片尺寸

#mousex -= mouse\_cursor.get\_width() / 2

#mousey -= mouse\_cursor.get\_height() / 2

#图片光标代替鼠标

screen.blit(mouse\_cursor, xy)

for key in sprites\_dict:#设置方向转换

sprite = sprites\_dict[key]

#若碰到边界则反向运动

if (sprite[0] + sprite[4])>xmax or sprite[0]<0:

sprite[2] = -sprite[2]

if (sprite[1] + sprite[5])>ymax or sprite[1]<0:

sprite[3] = -sprite[3]

#改变坐标位置

sprite[0] += (sprite[2])

sprite[1] += (sprite[3])

#绘出球的图

screen.blit(sprite[6], (sprite[0],sprite[1]))

sprite[7] = pygame.Rect((sprite[0], sprite[1]),(sprite[4],sprite[5]))

#若鼠标碰到球

if sprite[7].collidepoint(xy):

if key == 'star ball':#得到星星球加五分

scores += 5

catch\_star\_ball = 1

soundwav.play()

else:#若是其他球则游戏结束，初始化数据

game\_over()

level = 1

n\_ball = 1

seconds = 0

scores = 0

sprites\_dict = {'ball1':ball1\_list, 'star ball': ball5\_list}

pygame.time.set\_timer(USEREVENT, t)

break

#若碰到星星球，则重新生成一个星星球

if catch\_star\_ball:

del sprites\_dict['star ball']

catch\_star\_ball = 0

change\_ball(0, 0, n, 1)

#获取游戏状态

for e in pygame.event.get():

if e.type == QUIT:

exit()

if e.type == USEREVENT:

n\_ball += 1

level += 1

n += 10

if n > (xmax - 100 - xball\_begin) :

n = 0

pygame.time.set\_timer(USEREVENT, t)

tp\_ball = change\_ball(tp\_ball, n\_ball, n)

if e.type == USEREVENT+1:

scores += 1

seconds += 1

show\_level(level)

show\_time(seconds)

show\_scores(scores)

screen.blit(ball\_begin, ((100+ n),(200 + n/3)))

pygame.display.flip()

**三、数据结构与变量设置**

#窗口尺寸和画布

xmax = 640

ymax = 480

screen = pygame.display.set\_mode((xmax,ymax))

#计时器

clock = pygame.time.Clock()

#球的出发点

ball\_begin

########球和球的坐标########

#第一个球 黄球

ball1

xball1 = ball1.get\_height()

yball1 = ball1.get\_width()

#第二个球 绿球

ball2

xball2 = ball2.get\_height()

yball2 = ball2.get\_width()

#第三个球 橙球

ball3

xball3 = ball3.get\_height()

yball3 = ball3.get\_width()

#第四个球 蓝球

ball4

xball4 = ball4.get\_height()

yball4 = ball4.get\_width()

#星星球

ball5

xball5 = ball5.get\_height()

yball5 = ball5.get\_width()

########球的列表#######

#ball\_list [ x, y, step\_x, step\_y, height, width, surface, rect]

ball1\_list

ball2\_list

ball3\_list

ball4\_list

ball5\_list

######设置球的顺序（字典）

balls\_dict = {'1':ball1\_list, '2':ball2\_list, '3':ball3\_list, '4':ball4\_list, 'starball':ball5\_list}

######初始化游戏数据

n = 1#初始球

n\_ball = 1 #下一个球

tp\_ball = 1 #当前球

level = 1 #级别

t = 2000 #计时间隔

seconds = 0 #时间

scores = 0 #分数

catch\_star\_ball = 0 #是否碰到星星

**四、函数和关键算法描述**

**函数：**

#def函数定义

#状态量转为字符以便应用字典

def one\_num\_to\_str(n):

if n == 0:

return '0'

elif n == 1:

return '1'

elif n == 2:

return '2'

elif n == 3:

return '3'

elif n == 4:

return '4'

elif n == 5:

return '5'

elif n == 6:

return '6'

elif n == 7:

return '7'

elif n == 8:

return '8'

elif n == 9:

return '9'

#整型转换为字符以便应用字典

def int\_to\_str(n):

a = one\_num\_to\_str(n/1000) + one\_num\_to\_str((n/100)%10) + one\_num\_to\_str((n/10)%10) + one\_num\_to\_str(n%10)

return a

#转换球

def change\_ball(tp\_ball, n\_ball, n, special\_mode = 0):

a = 1#存储运动方向以及球编号的变化

b = 1#存储运动方向以及球编号的变化

if special\_mode:#判断是否是星星球

key = 'star ball'

key2 = 'star ball'

a = -a

if (seconds % 2):

b = -b

else:

key = one\_num\_to\_str(tp\_ball)#存储当前球

key2 = 'ball' + int\_to\_str(n\_ball)#存储下一个球

#设置字典中的球的各项数据，生成新球

sprites\_dict[key2] = [0,0,0,0,0,0,0,0]

sprites\_dict[key2][0] = (100 + n)

sprites\_dict[key2][1] = (200 + n/3)

sprites\_dict[key2][2] = balls\_dict[key][2] \* a

sprites\_dict[key2][3] = balls\_dict[key][3] \* b

sprites\_dict[key2][4] = balls\_dict[key][4]

sprites\_dict[key2][5] = balls\_dict[key][5]

sprites\_dict[key2][6] = balls\_dict[key][6]

sprites\_dict[key2][7] = pygame.Rect((100,200),(balls\_dict[key][4],balls\_dict[key][5]))

tp\_ball += 1

if tp\_ball == 5:#如果生成星星球，则下一个生成黄球

tp\_ball = 1

return tp\_ball

#游戏结束的逻辑

def game\_over():

#soundoverwav.play(1)

size = 48

font = pygame.font.Font(None, size)

text = font.render('Oops! GAME OVER!', 1, (240,128,128))

text\_pos = text.get\_rect(centerx = screen.get\_width()/2, centery = (screen.get\_height()/2 - size))

gameover.set\_alpha(180)#设置透明度

screen.blit(gameover,(0,0))

screen.blit(text, text\_pos)

show\_level(level)

show\_time(seconds)

show\_scores(scores)

pygame.display.flip()

h = 1

while h:

for e in pygame.event.get():

if e.type == QUIT:

exit()

if e.type == MOUSEBUTTONDOWN:#重新开始游戏

h = 0

#打印级别

def show\_level(level):

size = 36

font = pygame.font.Font(None, size)

text = font.render('level: ' + int\_to\_str(level), 1, (135,206,250))

screen.blit(text, ((xmax-165), 8))

#打印时间

def show\_time(seconds):

size = 36

font = pygame.font.Font(None, size)

text = font.render('time: ' + int\_to\_str(seconds), 1, (135,206,250))

screen.blit(text, ((xmax-165), 40))

#打印分数

def show\_scores(scores):

size = 36

font = pygame.font.Font(None, size)

text = font.render('scores: ' + int\_to\_str(scores), 1, (135,206,250))

screen.blit(text, ((xmax-165),72))

**关键算法：**

**1、生成新球和球的转换：**利用字典，把每个球的列表存入，按照顺序生成新球，对新球列表内的数据进行赋值，初始点在出发点，球的转换你按照编号++的顺序，若到第四个球则再从第一个球开始生成，星星球的生成是独立的，不参与普通球的生成循环。

**2、球的运动：**球的运动主要依靠球的列表x、y坐标的计算，x\_step、y\_step则是球每次（计时器间隔）运动的步数，按照初始方向直线移动，若碰到边框则方向变为负数，即反弹回去，与之前运动逻辑相同直至被鼠标碰到或再碰到边框。

**3、检测鼠标与球的碰撞：**每个球每次计算x、y坐标后即利用collidepoint(xy)函数检测与鼠标坐标的碰撞。

**4、贴图与音乐：**贴图主要使用pygame.image.load('image.bmp').convert()函数，并用ball\_begin.set\_colorkey((255,255,255))扣去背景色，必要时使用ball\_begin.set\_alpha()函数来设置透明度，音乐插入主要使用pygame.mixer.music类的函数。

**五、游戏效果实现**

游戏开始时的初始状态，黄色小飞机即鼠标



随着球增多，可捕捉星星球得分



碰到其他的球，游戏结束，点击鼠标可重新开始

