****

**广州南方学院**

**电气与计算机工程学院**

课程名称： python

实验题目： 猴子接香蕉

2022年 7月 2 日

**附：实验报告**

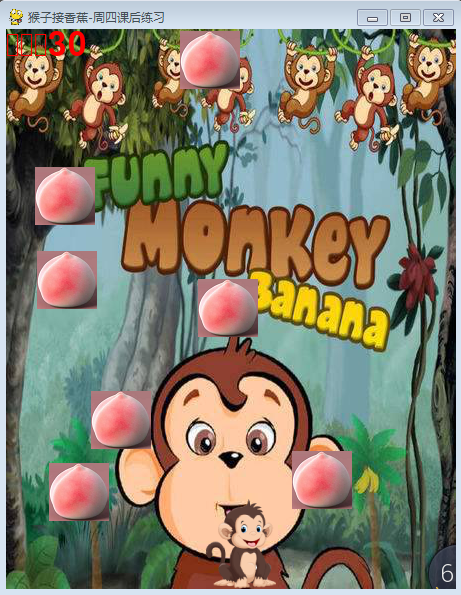
专业： 计算机科学与技术 年级： 2019 完成日期： 2022.7.2

学号： 1920170040 姓名： 周文博 成 绩：

**一、实验目的**

小猴子接香蕉游戏是一个根据游戏得分判定玩家反应力的游戏，该游戏的设定非常简单，游戏主体为香蕉和猴子：香蕉从屏幕顶端随机位置出现，垂直落下，玩家用鼠标左右键控制猴子左右移动，接住下落的香蕉，猴子每接到一个香蕉加10分。

本实例要求编写程序，实现一个小猴子接香蕉游戏。



**二、实验原理**

*（技术原理：文字、图、表、公式等）*

**三、实验过程**

（1）猴子接香蕉游戏可以分为4个部分，分别为加载游戏元素、设置猴子移动、香蕉移动与碰撞检测。

（2）首先需要绘制一个窗体，确保元素加载到该窗体中窗体的宽度可以设置为450，高度可以设置为560。

（3）然后加载背景图片、猴子图片、香蕉图片以及设置使用的字体和字号。

（4）定义方法，设置初始香蕉下落速度、分数、猴子位置、猴子移动速度变量。

（5）因为香蕉是随机从屏幕上方随机下落，所以我们可以定义一个列表来保存随机生成的香蕉坐标。

（6）因为是香蕉下落过程和猴子移动过程是一个连续移动的动画效果，所以需要创建Clock对象，控制图像的刷新频率。

（7）定义while循环，在while循环中渲染背景图像，并监听玩家输入的键盘事件。

（8）在猴子接香蕉游戏中，猴子仅支持向左移动和向右移动，也就是猴子只在X轴上进行移动，因此需要分别计算猴子左右移动后的坐标。

（9）在上述中已经明确猴子只能在X轴上左右移动，因此还需要进行边界判断，当猴子元素的X轴坐标大于窗体减去猴子图片宽度时，将猴子元素坐标设置为窗体宽度减去猴子图片宽度；当猴子元素X轴小于0时，将猴子元素X轴坐标设置为0，边界判断完成后将猴子元素绘制到窗体。

（10）猴子接香蕉中的香蕉是随机在窗体Y轴垂直下落，可使用for循环遍历已经准备好的香蕉坐标，为实现多个香蕉垂直下落效果，即香蕉Y轴坐标加指定的下落速度，若香蕉的Y轴坐标大于等于窗体高度时，则对香蕉图片元素的Y轴重新赋值保证可以香蕉图片可以再次循环下落，这里可以赋值为负的香蕉图片高度。

（11）在上述的for循环中已经实现香蕉的随机下落，接下来，就需要进行猴子与香蕉的碰撞检测，如果猴子与香蕉发生碰撞，那么进行分数累加。碰撞检测是根据香蕉下落坐标与猴子坐标进行判断，即判断X轴坐标与Y轴坐标是否重叠。

（12）与猴子发生碰撞香蕉可以应该执行“消失”操作。香蕉“消失”操作可以通过移除该香蕉的坐标实现。

（13）因为与猴子发生碰撞后的香蕉坐标被移除掉，所以当香蕉坐标列表中的元素为空时，不在有香蕉随机下落，为保证香蕉坐标列表元素永远不为空，可以判断香蕉坐标列表的长度，当长度小于指定长度时，将随机生成的坐标添加到香蕉坐标列表中。

（14）猴子每接到一个香蕉就会进行分数累加，该分数会在窗体的左上角显示，渲染分数文本并绘制到窗体上。

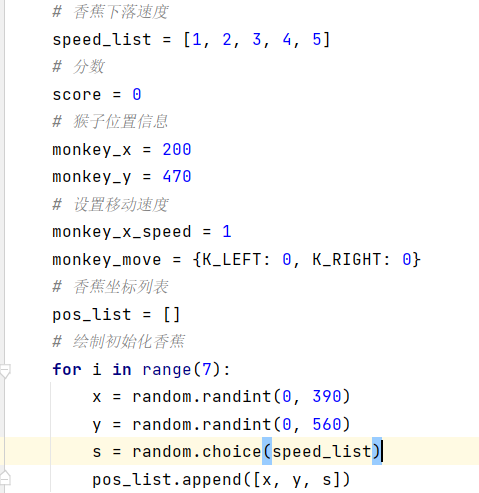
（15）最后更新屏幕显示内容

代码

**import** random  
**import** time  
**from** sys **import** exit  
**import** pygame  
**from** pygame.locals **import** \*  
  
pygame.init()  
*# 屏幕宽度*screen\_width = 450  
*# 屏幕高度*screen\_height = 560  
*# 游戏剩余时间*times = 10  
*# 绘制窗口*screen = pygame.display.set\_mode((screen\_width, screen\_height), 0, 32)  
  
*# 游戏标题*pygame.display.set\_caption(**"孙悟空接桃"**)  
*# 分数字体，字号*run\_time\_font = pygame.font.SysFont(**'simhei'**, 48)  
  
  
**def** game\_start():  
 *# 加载图片* monkey = pygame.image.load(**'monkey.png'**)  
 banana = pygame.image.load(**'tao1.png'**)  
 game\_background = pygame.image.load(**'background.jpg'**)  
 *# 香蕉下落速度* speed\_list = [1, 2, 3, 4, 5]  
 *# 分数* score = 0  
 *# 猴子位置信息* monkey\_x = 200  
 monkey\_y = 470  
 *# 设置移动速度* monkey\_x\_speed = 1  
 monkey\_move = {K\_LEFT: 0, K\_RIGHT: 0}  
 *# 香蕉坐标列表* pos\_list = []  
 *# 绘制初始化香蕉* **for** i **in** range(7):  
 x = random.randint(0, 390)  
 y = random.randint(0, 560)  
 s = random.choice(speed\_list)  
 pos\_list.append([x, y, s])  
 *# 帧率控制Clock对象* clock = pygame.time.Clock()  
 **while True**:  
 screen.blit(game\_background, (0, 0))  
 *# 接收信息处理* **for** event **in** pygame.event.get():  
 **if** event.type == QUIT:  
 exit()  
 **if** event.type == KEYDOWN:  
 **if** event.key **in** monkey\_move:  
 monkey\_move[event.key] = 1  
 **elif** event.type == KEYUP:  
 **if** event.key **in** monkey\_move:  
 monkey\_move[event.key] = 0  
 second\_time\_passed = clock.tick(60)  
 *# 定位猴子移动后坐标* monkey\_x -= monkey\_move[K\_LEFT] \* monkey\_x\_speed \* second\_time\_passed  
 monkey\_x += monkey\_move[K\_RIGHT] \* monkey\_x\_speed \* second\_time\_passed  
 *# 判断猴子边界条件* **if** monkey\_x > 450 - monkey.get\_width():  
 monkey\_x = 450 - monkey.get\_width()  
 **elif** monkey\_x < 0:  
 monkey\_x = 0  
 screen.blit(monkey, (monkey\_x, monkey\_y))  
 **for** y **in** pos\_list: *# 坐标循环，从y轴垂直下落* y[1] = y[1] + y[2]  
 screen.blit(banana, (y[0], y[1])) *# 绘制香蕉  
 # 没接到桃子检测* **if** y[1] >= 560:  
 score -= 10  
 y[1] = -banana.get\_height()  
 *# 碰撞检测* **if** monkey\_x < y[0] < monkey\_x + monkey.get\_width() **and** monkey\_y - banana.get\_height() < y[1] < monkey\_y:  
 score += 10  
 pos\_list.remove([y[0], y[1], y[2]])  
 x, y, s = random.randint(0, 390), random.randint(0, 560), random.choice(speed\_list)  
 **if** len(pos\_list) <= 6:  
 pos\_list.append([x, -y, s])  
  
 screen\_score = run\_time\_font.render(**'分数：'** + str(score), **True**, (255, 0, 0))  
 screen.blit(screen\_score, (0, 0))  
 screen\_time = run\_time\_font.render(**'剩余时间：'** + str(int(times - time.time() + start\_time)), **True**, (255, 0, 0))  
 screen.blit(screen\_time, (0, 50))  
 *# 读取最高记录* **try**:  
 **with** open(**r"record.txt"**, **"r"**) **as** f:  
 record = f.read()  
 **except** FileNotFoundError:  
 *# 初始化最高记录* **with** open(**r"record.txt"**, **"w+"**) **as** f:  
 record = 0  
 f.write(str(record))  
 screen\_score = run\_time\_font.render(**'最高记录：'** + str(record), **True**, (255, 0, 0))  
 screen.blit(screen\_score, (0, 100))  
  
 *# 刷新显示* pygame.display.update()  
 *# -100分游戏结束* **if** score == -100:  
 **return** -1  
 *# 游戏时间超过一分钟结束* **if** time.time() - start\_time >= times:  
 *# 刷新记录* **if** score > int(record):  
 **with** open(**r"record.txt"**, **"w+"**) **as** f:  
 f.write(str(score))  
 **return** -1  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 start\_time = time.time()  
 **while True**:  
 **if** game\_start() == -1:  
 **break** print(**"游戏结束"**)

**四、思考并回答以下问题**

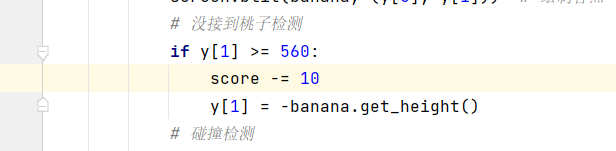
1. 将香蕉落下的速度随机，不固定。出现的位置也随机；

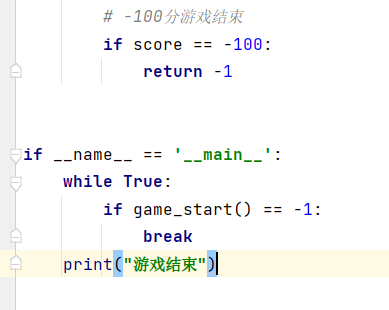


1. 换成孙悟空接桃；

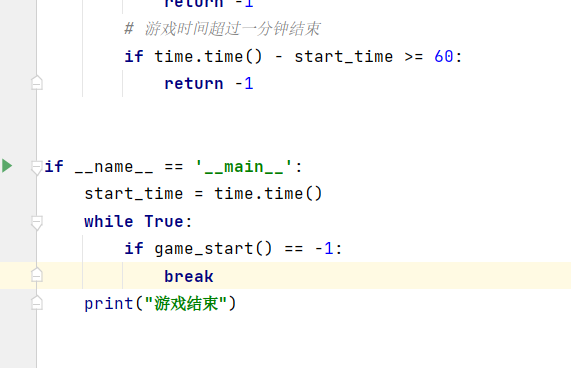


1. 加上惩罚机制，没接到会扣分，-100分游戏失败

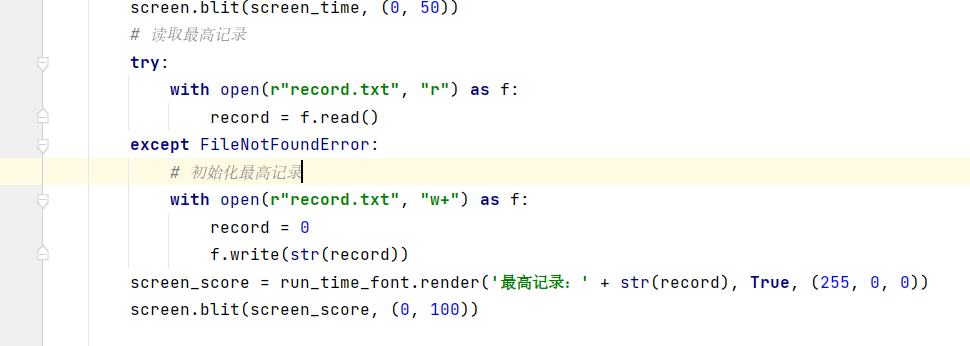


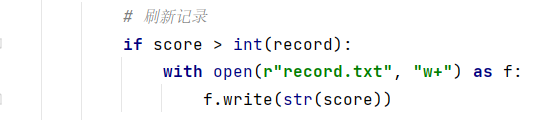


1. 更改游戏不是无限循环，游戏时间最多1分钟就结束



1. 加上最高分记录







1. 带有菜单重启等

很遗憾因为时间关系没能完成这个功能。

**五、实验总结和体会**

**在完成随机生成香蕉速度的时候，一开始我是随机生成speed变量，后来发现这个变量是每个香蕉都共同使用的，导致所有香蕉都是以同样的速度运动，并不是每个香蕉的运动速度不同，所以我在初始化香蕉的坐标时，把运动速度也初始化了，这样每个香蕉的运动速度才会不同。关于惩罚机制，分数到达-100分的时候，我就直接return -1退出函数了。计时是先记录开始时间，然后用当前时间减去开始时间。最高记录我是将分数保存到本地的txt文件，判断刷新了最高记录就把分数写进去，下次还能读取出来。最后一个菜单重启功能很遗憾实在没有时间写了，没能完成，等考完试有时间了我还会再研究研究，希望老师高抬贵手。**

**最后感谢老师一学期的教学，在这个学期里我学到了非常多的知识，感谢老师！**