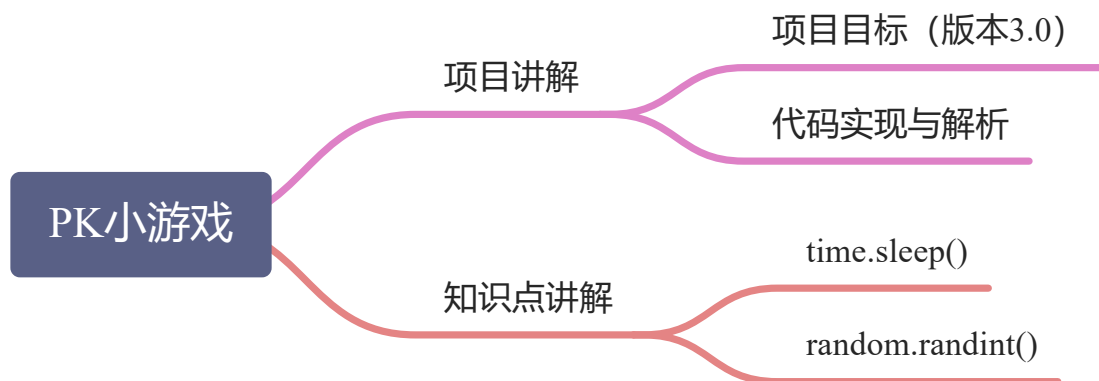


# 第7课 PK小游戏

## 一、课程结构导图



## 二、项目讲解

### 2.1 项目目标 (版本3.0)

自动生成玩家和敌人的属性，并且可以自动化对战过程，同时每局结束后显示对战结果，最终胜负采用三局两胜制。

### 2.2 代码实现及解析

```
1 import time, random
2
```

```
3 player_victory = 0
4 enemy_victory = 0
5
6 for i in range(1,4):
7     time.sleep(2) # 让局与局之间有较明显的有时间间隔
8     print(' \n—————现在是第'+str(i)+'局—————') # 作为局的标记
9
10    player_life = random.randint(100,150)
11    player_attack = random.randint(30,50)
12    enemy_life = random.randint(100,150)
13    enemy_attack = random.randint(30,50)
14
15    # 展示双方角色的属性
16    print('【玩家】 \n'+ '血量: '+str(player_life)+' \n攻击: '+str(player_attack))
17    print('-----')
18    time.sleep(1)
19    print('【敌人】 \n'+ '血量: '+str(enemy_life)+' \n攻击: '+str(enemy_attack))
20    print('-----')
21    time.sleep(1)
22
23    # 双方PK
24    while player_life > 0 and enemy_life > 0:
25        player_life = player_life - enemy_attack
26        enemy_life = enemy_life - player_attack
27        print('你发起了攻击,【敌人】剩余血量'+str(enemy_life))
28        print('敌人向你发起了攻击,【玩家】剩余血量'+str(player_life))
29        print('-----')
30        time.sleep(1.5)
31
32    #打印最终战果
33    if player_life > 0 and enemy_life <= 0:
34        player_victory += 1
35        print('敌人死翘翘了,你赢了! ')
36    elif player_life <= 0 and enemy_life > 0:
37        enemy_victory += 1
38        print('悲催,敌人把你干掉了! ')
39    else:
40        print('哎呀,你和敌人同归于尽了! ')
41
42    if player_victory > enemy_victory :
43        time.sleep(1)
44        print('【最终结果:你赢了!】')
45    elif enemy_victory > player_victory:
```

```
46     print('【最终结果：你输了！】')
47 else:
48     print('【最终结果：平局！】')
```

### 代码解析：

第一步：第10-21行由于需要随机生成双方的属性（属性为整数），共四个属性，包括敌我双方的生命值以及攻击力。所以我们需要使用random模块里的randint()函数。

第二步：24-30行将我们版本1.0里的战斗过程实现自动化，可以采用循环的方式，因为战斗的次数并不确定，所以选用while循环。只有双方的血量均大于0的时候，战斗才是继续的，只要有一方的血量小于等于0，战斗就会结束，所以while循环的循环条件应是双方的血量都大于0。

双方血量计算公式如下：【敌人】剩余血量=敌人当前血量-玩家攻击力，【玩家】剩余血量=玩家当前血量-敌人攻击力。

第三步：33-40行如果要打印战果，就需要理清楚玩家胜利，玩家失败和平局三种情况。

第四步：分解三局两胜为三局和两胜两个需求。如第6行三局我们可以用for i in range(3)：来做三次循环实现。

统计三局的胜利情况我们可以分别来统计每一次循环的胜利情况。为此我们在第3-4行可以设置player\_victory和enemy\_victory两个整数变量分别来表示玩家和敌人胜利的局数，并设置初始值为0，第33-40行每一局结束之后赢家所对应的变量值加一，并打印胜负。

第五步：42-48行当三局都结束之后我们通过判断player\_victory和enemy\_victory两个变量的大小来决定谁是最后的赢家。

## 三、知识点讲解

### 3.1 time.sleep()

**使用场景：**time是python中的一个模块，为了让终端中打印内容的显示有一定时间间隔，我们需要使用time库中的sleep()函数。

**用法：**我们需要先使用import time语句调用time模块。然后在停顿的语句之后输入time.sleep(sec)，暂停的时间就是sec的数值，单位为秒。

**示例：**

```
1 import time # 导入time模块
2 print('你好')
3 time.sleep(3) # 使程序暂停3秒
4 print('距离上一个你好已经过去了3秒')
```

## 3.2 random.randint()

**使用场景：**当我们需要计算机为我们生成一个随机的整数供我们操作的时候，我们可以使用random模块中的randint()函数。

**用法：**我们需要先使用import random语句调用random模块。然后用一个变量接收random.randint(a,b)返回的整数，a和b是生成整数的范围。

**示例：**

```
1 import random
2 number_guess = random.randint(1, 9)
3 count = 0 # 统计猜数次数
4 while True:
5     guess = int(input("请输入一个1-9之间的整数: "))
6     count += 1 # 次数加一
7     if guess > number_guess:
8         print("你猜的太大啦")
9     elif guess < number_guess:
10        print("你猜的太小啦")
11    else:
12        print("你用{}次猜对了结果".format(count))
13        break
```

