

# 上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

## 程序设计思想与方法

THE IDEAS AND METHODS OF  
COMPUTER PROGRAMMING

### 课程论文

THESIS OF CURRICULUM



论文题目： 关于 Python 大作业的设计报告

学生姓名及学号： 杨超琪 515030910067

学生姓名及学号： 吴志承 515030910076

学生姓名及学号： 詹典 515030910072

指导教师： \_\_\_\_\_ 蔡伟

学院(系)： 电子信息与电气工程学院



## 目录

摘要.....	1
Abstract.....	1
关键词.....	1
Key Words.....	1
1. 前言.....	2
1.1 背景.....	2
1.2 选题决定.....	2
2. 程序分析及设计.....	2
2.1 自上而下设计框架.....	2
2.2 具体代码及详细注释.....	3
3. 问题及解决方式.....	16
4. 测试与优化.....	18
4.1 测试人群.....	18
4.2 测试的结果.....	18
4.3 边界条件与报错.....	18
5. 工作量分配.....	18
6. 总结.....	18
7. 心得体会.....	19



# 关于 Python 大作业的设计报告

F1503003 杨超琪 吴志承 詹典

## 摘要

作为上海交通大学电子信息与电气工程学院的学生,程序设计无疑是我们的重中之重。在此程序设计的思想与方法课程结束之即,我们团队编写了一个小游戏----是蘑菇就下 100 层。本文主要介绍我们编写这个游戏的过程,我们遇到的困难,以及如何通过查阅资料来克服困难的经历,以及一些初尝编码的心得体会。

**关键词:** Python、游戏

## Abstract

As members of The School of Electrical Information and Electronic Engineering of SJTU, computer programming is absolutely important for us. By the end of the class of ideas and methods of computer programming, our group write a smart game called Jump For Fun ! This thesis mainly presents the procedure of creating this game, the problems we faced when working, and the experiences of searching for references to solve tough problems, as well as some feelings and thoughts as three freshmen.

**Key Word :** Python、computer games

## 1. 前言

### 1.1 背景

是蘑菇就下 100 层是一个网络经典的小游戏，网上的俗称叫：跳楼游戏、是男人就下 100 层，因为我们团队设计的这个程序主人公为一颗绿色的小蘑菇，所以我们又取名为：是蘑菇就下 100 层。

这个游戏的介绍如下：屏幕底部会随机出现长短不一的木板，慢慢往上运动，到了屏幕顶部就消失；主人公也就是那个蘑菇在 Y 轴方向上如果下方没有木板，那么会自动作自由落体运动，如果下方踩到木板，就会随着木板一起迅速往上运动，其 X 轴上的运动可由键盘上的“Left”和“right”键控制；屏幕上还会随机生成红色的“火焰木板”和“传送带木板”。

游戏中主人公有 3 条生命，失去生命的情况分为 3 种：1. 主人公头顶碰到天花板（也就是程序的屏幕顶部）；2. 主人公脚步碰到地狱（也就是屏幕的底部）；3. 主人公踩到“火焰木板”上。

### 1.2 选题决定

在制作这个游戏的决定产生前有一段曲折的里程。我组原先的决定是制作一个线性代数的计算器，在工作了一段时间之后，了解到制作线性代数计算器的小组比较多，顿感千篇一律，缺乏创新性，于是决定改变主题-----制作一个小游戏。我们想过贪吃蛇、五子棋、24 点计算方式分析、扫雷等等，最终根据编码能力与可玩性的考虑，选定是蘑菇就下 100 层。

## 2. 程序分析及设计

### 2.1 自上而下设计框架

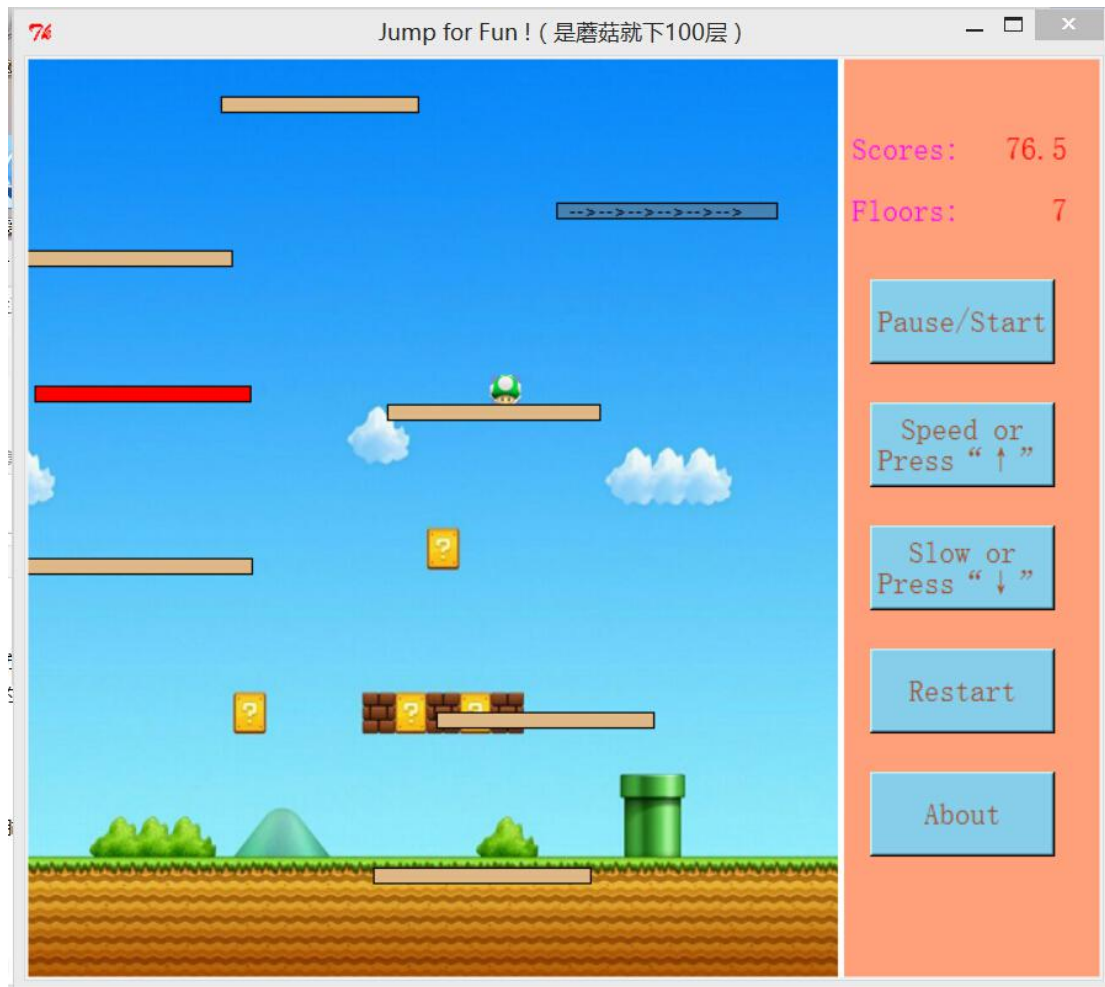
#### Step1. 设计游戏界面

- (1). 在草稿纸上粗略画出假想的界面样式。
- (2). 对照草图，详细列举需要处理的各个构件：
  - (a). 需要创建一个合适的 Tk 框，并创建界面 title。

- (b). 根据 Tk 框的大小创建合适数量的等间距随机长度随机位置的木板。
- (c). 从网上找到合适的卡通人偶图片（蘑菇）插入 Tk 框。
- (d). 创建红色和黄色等间距随机长度随机位置的木板分别充当“火焰木板”和“传送带木板”。
- (e). 从网上找到合适的图片当做背景图片。
- (f). 创建竖线分割显示栏和功能栏。
- (g). 在功能栏分别创建 Score、Pause/Start、Speed up、Slow down、Restart、About 按钮。

(3). 具体确定或计算各个构件所处的位置信息，以及其分别的大小信息。

### Step2. 分别为各个构件编码实现其功能



## 2.2 具体代码及详细注释

```
#coding:utf-8
from Tkinter import*
from random import*
from time import*
import Image
#导入所需库
root=Tk()
root.geometry("700x600")
root.title("Jump for Fun ! (是蘑菇就下100层)")
#创建Tk界面
k,click,1,height=1,0,0,12
X,x=260,185
score=0
over=0
Floors=0
dm=0
#变量初始化 k 是木板移动速度
#click 是用于控制暂停和运行的一个变量
#1 是用于随机变化传送带方向的变量
#height 是用于控制木板长度的变量
#X 用于表示小蘑菇的纵坐标
#x 是用于表示小蘑菇的横坐标
#over 用于 restart 函数的
#Floors 是个计数器，计算下落的层数
#dm 是计数器的过程变量

pic=PhotoImage(file='C:/picture2.ppm')
pic2=PhotoImage(file='C:/picture1.ppm')
#导入了背景图片和小蘑菇

def pause():
    global k
    global l
    global click
    click=click+1
    if click%2==1:
        l=k
        k=0
    else:
        k=1
#暂停和运行功能
def Speed(event):
```



```
global k
if k<3 and event.keysym == 'Up' :
    k=k+0.3
#加速功能和 Up 兼容
def Slow(event):
    global k
    if k>0.3 and event.keysym == 'Down':
        k=k-0.3
#减速功能和 Down 兼容
def about():
    top=Toplevel()
    canvas=Canvas(top,width=500,height=400,bg='black')
    canvas.pack()
    Label(canvas,text="OUR TEAM\n \nTeamLeader: Chaoqi Yang ( SJTU ) \nAcadamy:
SEIEE\nCellphone: 15\
8-2118-8556\nE-mail: Never_say_never@sjtu.edu.cn\n\nTeamMember1: Dian Zhan (SJTU)
\nAcadamy:\
SEIEE\nCellphone: 131-6209-6959\nE-mail: 616042024@qq.com\n\nTeamMember2:
Zhicheng Wu (SJTU) \n\
Acadamy: SEIEE\nCellphone: xxx-xxxx-xxxx\nE-mail:
1013041712@qq.com",fg='white',bg='black',font=30,justify='left').pack()
    top.title("Designer-ROYAL CLUB")
    if click%2==0:
        pause()
#用于显示组员信息，提升逼格
def moveplayer(event):
    global x
    global g
    global x8
    global width8
    global over
    if g-height<=X<=g and (x+15>x8 and x<x8+width8-3):
        pass
    elif over!=1:
        if event.keysym == 'Left' and x>4*(k+1):
            cn.move(player,-4*(k+1),0)
            x=x-4*(k+1)
        elif event.keysym == 'Right' and x<512-4*(k+1):
            cn.move(player,4*(k+1),0)
            x=x+4*(k+1)
    else:
        pass
#用键盘的 right 与 left 键控制小蘑菇的左右运动
def restart():#创建重新开始游戏的功能
```

```
global r1
global r2
global r3
global r4
global r5
global r6
global r7
global r8
global r9
global player
global label
global pic
global over
over=0
Floors=0
cn.delete(r1)
cn.delete(r2)
cn.delete(r3)
cn.delete(r4)
cn.delete(r5)
cn.delete(r6)
cn.delete(r7)
cn.delete(r8)
cn.delete(r9)
cn.delete(player)
label.destroy()#这边把原先那一盘的界面摧毁，防止新建构建与其重叠
global X
global x
global k
global score
global click
global height#因为此处涉及各种变量，于是直接无脑全部 global
k,click,1,height=1,0,0,10#这里初始化各种变量
X,x=260,185#这个是创建小蘑菇的初始位置
score=0#初始化分数
Label(cn,text='          ',bg='blue').place(x=680,y=60,anchor=E)

cn.bind_all('<KeyPress-Left>',moveplayer)
cn.bind_all('<KeyPress-Right>',moveplayer)
cn.bind_all('<KeyPress-Up>',Speed)
cn.bind_all('<KeyPress-Down>',Slow)#将按键与功能 bind 起来

cn.update()
a,b,c,d,e,f,h,g=500,400,300,200,100,600,550,50
```

```

x1,width1=draw()
r1=cn.create_rectangle(x1,500-height,x1+width1,500,fill='BurlyWood')
x2,width2=draw()
r2=cn.create_rectangle(x2,400-height,x2+width2,400,fill='BurlyWood')
x3,width3=draw()
r3=cn.create_rectangle(x3,300-height,x3+width3,300,fill='BurlyWood')
x4,width4=draw()
r4=cn.create_rectangle(x4,200-height,x4+width4,200,fill='BurlyWood')
x5,width5=draw()
r5=cn.create_rectangle(x5,100-height,x5+width5,100,fill='BurlyWood')
x6,width6=draw()
r6=cn.create_rectangle(x6,600-height,x6+width6,600,fill='BurlyWood')
player=cn.create_image(195,250,image=pic)#重建6块木板

x7,width7=draw()
r7=cn.create_rectangle(x7,550-height,x7+width7,550,fill='red')#重建火焰木板
x8,width8,l=drawline()
r8=cn.create_rectangle(x8,50-height,x8+width8,50,fill='SteelBlue')#重建传送带
if l==1:
    r9=cn.create_text(x8,50-height/2,text=' -->-->-->-->-->--> ',anchor=W)
else:
    r9=cn.create_text(x8,50-height/2,text=' <--<--<--<--<--<--< ',anchor=W)
#用于随机变化传送带的方向
Label(cn,text='
',font=30,bg='LightSalmon',fg='red').place(x=680,y=60,anchor=E)
q=cn.create_oval(254,239,326,316,width=7,fill='black',outline='purple')
for i in ('3','2','1','Go','1'):
    labelq=Label(cn,text=i,font=(30,30),bg='black',fg='white')
    labelq.place(x=291,y=299,anchor=S)
    sleep(1)
    root.update()
    labelq.destroy()
cn.delete(q)#创建倒计时牌
while True:
    global k,Floors,dm

    if (X<a-height or X>a) and (X<b-height or X>b) and (X<c-height or X>c) and
(X<d-height or X>d) \
    and (X<e-height or X>e) and (X<f-height or X>f)and (X<g-height or X>g):
        cn.move(player,0,1.8*k)
        X=X+1.8*k#自然下落的速度
        dm+=2.8*k#计数楼层的过程变量，用了相对速度

```

```

elif g-height<=X<=g and (x+15>x8 and x<x8+width8-3):
    cn.move(player,0,-k*1.5)
    X=X-k*1.5
    if l==1:
        cn.move(player,k,0)
        x=x+k
    else:
        cn.move(player,-k,0)
        x=x-k#检验是否在传送带上
elif a-height<=X<=a and (x+15<=x1 or x>=x1+width1-3):
    cn.move(player,0,1.8*k )
    X=X+1.8*k
    dm+=2.8*k
elif b-height<=X<=b and (x+15<=x2 or x>=x2+width2-3):
    cn.move(player,0,1.8*k )
    X=X+1.8*k
    dm+=2.8*k
elif c-height<=X<=c and (x+15<=x3 or x>=x3+width3-3):
    cn.move(player,0,1.8*k )
    X=X+1.8*k
    dm+=2.8*k
elif d-height<=X<=d and (x+15<=x4 or x>=x4+width4-3):
    cn.move(player,0,1.8*k )
    X=X+1.8*k
    dm+=2.8*k
elif e-height<=X<=e and (x+15<=x5 or x>=x5+width5-3):
    cn.move(player,0,1.8*k )
    X=X+1.8*k
    dm+=2.8*k
elif f-height<=X<=f and (x+15<=x6 or x>=x6+width6-3):
    cn.move(player,0,1.8*k)
    X=X+1.8*k
    dm+=2.8*k#检验是否在 6 块木板上
elif a-height<=X<=a and (x+15>x1 and x<x1+width1-3) or\
    b-height<=X<=b and (x+15>x2 and x<x2+width2-3) or\
    c-height<=X<=c and (x+15>x3 and x<x3+width3-3) or\
    d-height<=X<=d and (x+15>x4 and x<x4+width4-3) or\
    e-height<=X<=e and (x+15>x5 and x<x5+width5-3) or\
    f-height<=X<=f and (x+15>x6 and x<x6+width6-3) :
    cn.move(player,0,-k)
    X=X-k
else :
    cn.move(player,0,1.8*k)
    X=X+1.8*k

```

```
dm+=2.8*k

if dm>100:
    Floors+=1
    dm=dm-100#变量初始化

a=a-k
cn.move(r1,0,-k)
if a<0:
    cn.delete(r1)
    x1,width1=draw()
    r1=cn.create_rectangle(x1,600-height,x1+width1,600,fill='BurlyWood')
    a=600
#控制木板 a 的运动
b=b-k
cn.move(r2,0,-k)
if b<0:
    cn.delete(r2)
    x2,width2=draw()
    r2=cn.create_rectangle(x2,600-height,x2+width2,600,fill='BurlyWood')
    b=600
#控制木板 b 的运动
c=c-k
cn.move(r3,0,-k)
if c<0:
    cn.delete(r3)
    x3,width3=draw()
    r3=cn.create_rectangle(x3,600-height,x3+width3,600,fill='BurlyWood')
    c=600
#控制木板 c 的运动
d=d-k
cn.move(r4,0,-k)
if d<0:
    cn.delete(r4)
    x4,width4=draw()
    r4=cn.create_rectangle(x4,600-height,x4+width4,600,fill='BurlyWood')
    d=600
#控制木板 d 的运动
e=e-k
cn.move(r5,0,-k)
if e<0:
    cn.delete(r5)
    x5,width5=draw()
    r5=cn.create_rectangle(x5,600-height,x5+width5,600,fill='BurlyWood')
```

```

        e=600
#控制木板 e 的运动
        f=f-k
        cn.move(r6,0,-k)
        if f<0:
            cn.delete(r6)
            x6,width6=draw()
            r6=cn.create_rectangle(x6,600-height,x6+width6,600,fill='BurlyWood')
            f=600
#控制木板 f 的运动
        h=h-2*k
        cn.move(r7,0,-2*k)
        if h<0:
            cn.delete(r7)
            x7,width7=draw()
            r7=cn.create_rectangle(x7,600-height,x7+width7,600,fill='red')
            h=600
#控制木板 h 的运动
        g=g-k*1.5
        cn.move(r8,0,-k*1.5)
        cn.move(r9,0,-k*1.5)
        if g<0:
            cn.delete(r8)
            cn.delete(r9)
            x8,width8,l=drawline()
            r8=cn.create_rectangle(x8,600-height,x8+width8,600,fill='SteelBlue')
            if l==1:
                r9=cn.create_text(x8,600-height/2,text='                -->-->-->-->-->-->'
,anchor=W)
            else:
                r9=cn.create_text(x8,600-height/2,text='                <--<--<--<--<--<'
,anchor=W)
            g=600
#控制传送带的运动

        if X<20 or X>600 or (h-height<=X<=h and x7<=x<=x7+width7):
            label=Label(cn,text=' Game Over !!!',font=30,bg='MidnightBlue',fg='red')
            label.place(x=180,y=300,anchor=W)
            over=1#game over 的出现
            break
        score+=k*0.1#分数计数
        Label(cn,text='
'+str(score),font=30,bg='LightSalmon',fg='red').place(x=680,y=60,anchor=E)

```

```
Label(cn, text='
'+str(Floors), font=30, bg='LightSalmon', fg='red').place(x=680, y=100, anchor=E)
    sleep(0.01)
    root.update()#更新
def restart2():
    global over
    if over==1 :
        restart()#控制 restart 函数

cn=Canvas(root, width=700, height=600, bg='LightSalmon')
cn.pack()

#创建画布

cn.create_image(265, 300, image=pic2)
b1=Button(cn, text="Pause/Start", height=2, width=11, command=pause, bg="SkyBlue", fg="Sienna", font=30)
b1.place(x=669, y=200, anchor=SE)
b2=Button(cn, text="Speed          or\nPress          “          ↑          ”
", height=2, width=11, command=Speed, bg="SkyBlue", fg="Sienna", font=30)
b2.place(x=669, y=280, anchor=SE)
b3=Button(cn, text="Slow          or\nPress          “          ↓          ”
", height=2, width=11, command=Slow, bg="SkyBlue", fg="Sienna", font=30)
b3.place(x=669, y=360, anchor=SE)
b4=Button(cn, text="Restart", height=2, width=11, command=restart2, bg="SkyBlue", fg="Sienna", font=30)
b4.place(x=669, y=440, anchor=SE)
b5=Button(cn, text="About", height=2, width=11, command=about, bg="SkyBlue", fg="Sienna", font=30)
b5.place(x=669, y=520, anchor=SE)

a2=Label(cn, text="Scores:
", bg='LightSalmon', fg='Magenta', font=30, width=15, height=2, anchor=W)
a2.place(x=690, y=60, anchor=E)
a3=Label(cn, text="Floors:
", bg='LightSalmon', fg='Magenta', font=30, width=15, height=2, anchor=W)
a3.place(x=690, y=100, anchor=E)

cn.create_line(530, 0, 530, 600, width=4, fill='white')
#创建各种按钮
cn.create_text(620, 560, text='PS：不要碰到红色木板；\n碰到蓝色木板会按箭头移动；
\n左右键控制蘑菇左右移动；\n上下键控制下落加速减速。', fill='red')
#创建提示板
cn.bind_all('<KeyPress-Left>', moveplayer)
```

```
cn.bind_all('<KeyPress-Right>',moveplayer)
cn.bind_all('<KeyPress-Up>',Speed)
cn.bind_all('<KeyPress-Down>',Slow)

def draw():
    ran1=uniform(120,150)
    ran2=uniform(0,520)
    ran4=uniform(0,10)
    if ran1+ran2<=530:
        position=ran2
    elif ran4>=5:
        position=526-ran1
    else:
        position=0
    return position,ran1
#随机函数控制木板的长度，位置
def drawline():
    ran1=uniform(120,150)
    ran2=uniform(0,520)
    ran4=uniform(0,10)
    ran5=uniform(0,10)
    if ran1+ran2<=530:
        position=ran2
    elif ran4>=5:
        position=526-ran1
    else:
        position=0
    if ran5>=5:
        q=1
    else:
        q=-1
    return position,ran1,q
#也用于木板的随机量控制
a,b,c,d,e,f,h,g=500,400,300,200,100,600,550,50
x1,width1=draw()
r1=cn.create_rectangle(x1,500-height,x1+width1,500,fill='BurlyWood')
x2,width2=draw()
r2=cn.create_rectangle(x2,400-height,x2+width2,400,fill='BurlyWood')
x3,width3=draw()
r3=cn.create_rectangle(x3,300-height,x3+width3,300,fill='BurlyWood')
x4,width4=draw()
r4=cn.create_rectangle(x4,200-height,x4+width4,200,fill='BurlyWood')
x5,width5=draw()
r5=cn.create_rectangle(x5,100-height,x5+width5,100,fill='BurlyWood')
```



```

x6,width6=draw()
r6=cn.create_rectangle(x6,600-height,x6+width6,600,fill='BurlyWood')

x7,width7=draw()
r7=cn.create_rectangle(x7,550-height,x7+width7,550,fill='red')
x8,width8,l=drawline()
r8=cn.create_rectangle(x8,50-height,x8+width8,50,fill='SteelBlue')
if l==1:
    r9=cn.create_text(x8,50-height/2,text=' -->-->-->-->-->--> ',anchor=W)
else:
    r9=cn.create_text(x8,50-height/2,text=' <--<--<--<--<--<--< ',anchor=W)
player=cn.create_image(195,250,image=pic)
#第一轮创建各种构建
Label(cn,text='
',font=30,bg='LightSalmon',fg='red').place(x=680,y=60,anchor=E)
q=cn.create_oval(254,239,326,316,width=7,fill='black',outline='purple')
for i in ('3','2','1','Go','1'):
    labelq=Label(cn,text=i,font=(30,30),bg='black',fg='white')
    labelq.place(x=291,y=299,anchor=S)
    sleep(1)
    root.update()
    labelq.destroy()
cn.delete(q)#创建倒计时牌
while True:
    global k,Floors,dm
    global x
    global X
    global score

    if (X<a-height or X>a) and (X<b-height or X>b) and (X<c-height or X>c) and
(X<d-height or X>d) \
        and (X<e-height or X>e) and (X<f-height or X>f)and (X<g-height or X>g):
        cn.move(player,0,1.8*k)
        X=X+1.8*k
        dm+=2.8*k
    elif g-height<=X<=g and (x+15>x8 and x<x8+width8-3):
        cn.move(player,0,-k*1.5)
        X=X-k*1.5
        if l==1:
            cn.move(player,k,0)
            x=x+k

```

```
    else:
        cn.move(player, -k, 0)
        x=x-k
    elif a-height<=X<=a and (x+15<=x1 or x>=x1+width1-3):
        cn.move(player, 0, 1.8*k )
        X=X+1.8*k
        dm+=2.8*k
    elif b-height<=X<=b and (x+15<=x2 or x>=x2+width2-3):
        cn.move(player, 0, 1.8*k )
        X=X+1.8*k
        dm+=2.8*k
    elif c-height<=X<=c and (x+15<=x3 or x>=x3+width3-3):
        cn.move(player, 0, 1.8*k )
        X=X+1.8*k
        dm+=2.8*k
    elif d-height<=X<=d and (x+15<=x4 or x>=x4+width4-3):
        cn.move(player, 0, 1.8*k )
        X=X+1.8*k
        dm+=2.8*k
    elif e-height<=X<=e and (x+15<=x5 or x>=x5+width5-3):
        cn.move(player, 0, 1.8*k )
        X=X+1.8*k
        dm+=2.8*k
    elif f-height<=X<=f and (x+15<=x6 or x>=x6+width6-3):
        cn.move(player, 0, 1.8*k)
        X=X+1.8*k
        dm+=2.8*k
    elif a-height<=X<=a and (x+15>x1 and x<x1+width1-3) or\
        b-height<=X<=b and (x+15>x2 and x<x2+width2-3) or\
        c-height<=X<=c and (x+15>x3 and x<x3+width3-3) or\
        d-height<=X<=d and (x+15>x4 and x<x4+width4-3) or\
        e-height<=X<=e and (x+15>x5 and x<x5+width5-3) or\
        f-height<=X<=f and (x+15>x6 and x<x6+width6-3) :
        cn.move(player, 0, -k)
        X=X-k
    else :
        cn.move(player, 0, 1.8*k)
        X=X+k*1.8
        dm+=2.8*k
    if dm>=100:
        Floors+=1
        dm=dm-100
```

```
a=a-k
cn.move(r1,0,-k)
if a<0:
    cn.delete(r1)
    x1,width1=draw()
    r1=cn.create_rectangle(x1,600-height,x1+width1,600,fill='BurlyWood')
    a=600

b=b-k
cn.move(r2,0,-k)
if b<0:
    cn.delete(r2)
    x2,width2=draw()
    r2=cn.create_rectangle(x2,600-height,x2+width2,600,fill='BurlyWood')
    b=600

c=c-k
cn.move(r3,0,-k)
if c<0:
    cn.delete(r3)
    x3,width3=draw()
    r3=cn.create_rectangle(x3,600-height,x3+width3,600,fill='BurlyWood')
    c=600

d=d-k
cn.move(r4,0,-k)
if d<0:
    cn.delete(r4)
    x4,width4=draw()
    r4=cn.create_rectangle(x4,600-height,x4+width4,600,fill='BurlyWood')
    d=600

e=e-k
cn.move(r5,0,-k)
if e<0:
    cn.delete(r5)
    x5,width5=draw()
    r5=cn.create_rectangle(x5,600-height,x5+width5,600,fill='BurlyWood')
    e=600

f=f-k
cn.move(r6,0,-k)
if f<0:
    cn.delete(r6)
```

```

        x6,width6=draw()
        r6=cn.create_rectangle(x6,600-height,x6+width6,600,fill='BurlyWood')
        f=600

    h=h-2*k
    cn.move(r7,0,-2*k)
    if h<0:
        cn.delete(r7)
        x7,width7=draw()
        r7=cn.create_rectangle(x7,600-height,x7+width7,600,fill='red')
        h=600

    g=g-k*1.5
    cn.move(r8,0,-k*1.5)
    cn.move(r9,0,-k*1.5)
    if g<0:
        cn.delete(r8)
        cn.delete(r9)
        x8,width8,l=drawline()
        r8=cn.create_rectangle(x8,600-height,x8+width8,600,fill='SteelBlue')
        if l==1:
            r9=cn.create_text(x8,600-height/2,text='                -->-->-->-->-->-->'
',anchor=W)
        else:
            r9=cn.create_text(x8,600-height/2,text='                <--<--<--<--<--<'
',anchor=W)
        g=600

    score+=k*0.1

Label(cn,text=str(score),font=30,bg='LightSalmon',fg='red').place(x=680,y=60,an
chor=E)

Label(cn,text=str(Floors),font=30,bg='LightSalmon',fg='red').place(x=680,y=100,
anchor=E)

    if X<20 or X>600 or (h-height<=X<=h and x7<=x<=x7+width7):
        label=Label(cn,text=' Game Over !!!',font=30,bg='MidnightBlue',fg='red')
        label.place(x=180,y=300,anchor=W)
        over=1
        break
    sleep(0.01)
    root.update()

```

#大概分成两段，前面一段是个函数，用于 restart () ，后面一段是第一盘游戏的设置

#这两段里面的内容其实是差不多的，为什么我们没有把两段合在一起呢？因为原先创建的  
#时候就是这个样子，如果要改掉的话感觉好麻烦，就这样吧，因为把第一盘结束后和第二  
盘  
#开始前接起来的程序有点烧脑，于是就没去考虑了。

```
root.mainloop()
```

### 3. 问题及解决方式

**Q1:**在插入背景图片与卡通图片时，在 Tk 界面上始终显示不出来？

**A1:**通过课本 P182 下方的注解从网上下载了 Python 图像库 PIL，并百度到了 PIL 的基本用法。

**Q2:**图片只能显示长方形的样式，用 Photoshop 等软件修改无用。

**A1:**先通过问学长学姐，然而并没有结果。然后通过百度，发现了几种修改图片样式的方式，并仔细钻研其代码意义，然后成功修改了图片，但是在导入到 Python 的 Tk 界面时发现并没有做修改。那天我们从晚上 9 点做到凌晨 2 点，一直在尝试着修复这个问题，最后在网上看到一个帖子说 alpha 用法通道不能作用于 Tk 界面，于是我们尝试着在程序中获取背景的 pixel 并将其赋值给主人公图片的白色边框，但是代码修改了好多次，不但没有作用（主人公图片不见了），而且程序运行速度变慢了，我们就放弃了。最后采用组员 member1 詹典的想法，将白色边框的 RGB 值改成了背景色相似的天蓝色。

**Q3:**当我将程序发给许多同学测试的时候，他们给我的回馈是既然名字叫“是蘑

菇就下 100 层”，那么就必须要有一个计数器（计算下的层数）。

**A3:** 这个问题比较容易解决。虽然一开始我想当然得把计数器的计数速度设成小蘑菇的下落速度，然而在测试的时候发现并不是下一层计数器加 1，我就详细地检查了一遍计数的步骤，发现没错；于是我又以为是程序运行系统误差，但是在室友的电脑上跑起来计数都是有偏差的，又感觉不是程序误差，而是自己编码错误。最后经过我们小组讨论，详细地分析了计数的过程，最后发现是个物理问题，竟然要考虑到相对速度！！！恍然大悟。

**Q4:** 当编写好程序自己测试时，当时是通过按钮来加速和减速的，发现用鼠标带点 Speed up 和 Slow down 时，游戏很容易 game over 。

**A4:** 于是我们又加上了通过用 Up 键和 Down 键和按钮同时控制加速减速的功能，利用 key-press 功能。

**Q5:** 在程序运行一段时间后，程序有时候会变卡。

**A5:** 百度了一下这个情况，网上有人说：1. 可以优化程序；2. 可以用 exe 打包在 windows 命令行 cmd 里面运行；3. 利用 pypy 或者 psyco 等软件进行外部包装。首先我们先去网上学习了一下 py2exe 的打包功能，将我们的程序进行了打包，然后又重新浏览了一遍程序，将其优化，删除了一些不必要的代码，运行速度快了不少。然而运行长时间后，仍然会出现类似的问题；然后我们又在网上学习 pypy 与 psyco 的用法，但是 psyco 下载了好多次，安装始终不成功，发现是只支持 python2.6 以下的版本，我们又去下载 pypy，但是发现安装包里只有一个类似 python 命令行的 exe 文件，没有 IDLE 文件，然后，在这个问题上宣告放弃。

**Q6:** 在游戏操作上，具有一些可玩性的问题，比如：蘑菇下降速度与木板上移动速度不匹配、蘑菇有时候会略微被木板遮住。

**A6:** 第一个问题困扰了我们好长时间，最有效的方法就是去 4399 小游戏网站上玩一下专业人士编的跳楼游戏，然后感性地修改移动速度。第二个问题是最令人头疼的，我们想了很多方法还是无法改正过来，因为我编的程序里小蘑菇和木板的移动都是像素化的，是一些离散的点，用 sleep 函数让其每秒帧数增多通过视觉延迟来达到连续移动效果的。因此很难控制小蘑菇每次都能顺利刚好落在木板上，总是要差一点点，如果速度加上去超过了极限，有时小蘑菇甚至会直接穿过木板！

于是，我又加了个速度限制。

还有诸多细小的问题在编写的时候困扰了我好久好久，在这里不再一一列举。

## 4. 测试与优化

### 4.1 测试人群

年龄在 17-20 岁的来自全国各高校的部分学生（所有高中同学）

### 4.2 测试的结果

在我们自己测试完发给同学后，总体来讲同学们的反映还是不错的，还有一些同学给我提了一些建议，比如：加上下落的层数、将加速减速改成用 Up 键和 Down 键控制等等。

### 4.3 边界条件与报错

在一遍又一遍的测试过程中，我们几乎把所有 bug 都解决掉了，因此，此程序应该是没有什么异常错误了。关于边界条件，这个程序与边界条件有关的应该是速度的限制和蘑菇的左右自动限制，然而这两方面我已经在程序中用代码解决了，所以应该边界条件也没有什么问题了的。

虽然常听老师讲测试优化的过程比编写程序来的更麻烦，更复杂，而且我们 debug 与测试优化的时间确实比写代码的时间长好多。但是因为我们是初学者，写完代码成就感远远大于 debug 的成就感，于是测试优化的过程感受与印象也不深了，就写到这里吧。

## 5. 工作量分配

组长杨超琪：提出游戏的设想，完成大部分编码与 debug 工作，完成大部分查阅

网络文档的任务，并且完成了大作业报告。

组员詹典：因为对于色彩比较敏感，修改 TK 界面的颜色与样式，使之搭配更和

谐，并且修改了大作业报告。并且提供了程序设计的大体框架，提出

新颖的想法，增加游戏可玩性。

组员吴志承：作为程序后期测试环节的重要成员，提出可以增加生命值的想法（虽

然最后并没有采用），修改了游戏中蘑菇的运行速度，使之与木板

的运行速度相符合，增加游戏可玩性。



## 6. 总结

这次大作业真的教会了我们很多课本上学不到的知识，也正如蔡伟老师所说的那样，可能是因为课本专注于讲解编程的思想而忽略了一些编程的实际细节吧。在大作业编写过程中需要用到许多课本上没有讲到或者一笔带过得知识，比如：Label 函数字体有 font 大小功能，键盘互动 Key-Press 功能等等，也正因为课本知识点不细致，我们才有机会为了一个小小的程序细节而浏览几十个网页，才有机会反对教条主义，为了自己热爱的程序而疯狂地在网页上找所需要的营养食粮。通过大作业的编写，我们几乎把设计到的构件的全部功能都用过一遍了，我想这才是大作业的真正意义----发现问题，自主解决，翻阅文档，提高能力。

## 7. 心得体会

**杨超琪**：大作业，虽然也被称为“作业”，但是前面加上一个“大”字就有了完全不同的意义。这个东西我想就是大学与高中的区别吧，项目没有任何限制，完全主动，投入自己的兴趣自主选题，自主研发，作出一个成果。

在大作业编写的过程中，让我体会到了大学与高中的区别，体会到了小组合作的快乐。小组成员一起为了实现一个成果而时常加班加点，都投入自己的热情，在完成成果的过程中体验到了满满的成就感，发现原先以为是很神秘高端的各种软件代码自己也能写出来。

经过这次大作业的考验，我觉得要写好代码的不二法门就是---多练多打，练习是提升能力最有效的方式。第二，团队合作非常重要，每个人都有不同的思维方式，每个人都有各自的 idea，只有把把家的 idea 结合起来才能做出最好的成果。

**詹典：**关于这次程序设计的大作业，经过我们讨论之后，组长完成了程序的关键函数。组长用一天时间写完了程序的第一版，并成功运行。但是，制作一个游戏并没有我们想的那么简单，初版存在各种各样的问题，主要是功能太简陋（如下落速度太慢、加速减速点击不方便、开始时缺少准备时间等）和界面外观不美观（色调单一暗淡、背景不搭配）。之后一星期我们都从各个方面对程序进行改进，为游戏增加难度，增添层数计数器，增加 321GO 倒计时，在改进过程中，我们的工具只有百度（有点无力），然后随着改进和优化，各种各样的 BUG 接踵而至，边界问题也需要考虑得更完备，改进似乎一直没有结束的预兆。直到 DEADLINE 接近，我们终于把这个游戏完成的差不多，界面明亮鲜丽，加速减速方便，游戏开始前有了倒计时，分数下有了层数的记录，蘑菇下落速度加快。游戏中还存在小问题，如程序的卡顿，图片边界的透明化难以实现，我们暂时没能找到解决方案。

经过这次大作业，我认识到一个优秀的程序除了最初的编写之外，后期的测试与改进同样重要，甚至工作量更大。为了让程序能更加完美，我们耗费了很多的精力和时间。此外，团队合作也很重要，前期工作主要是组长完成，略显艰难，后来我们小组共同讨论分工完成了不少工作。总之，通过大作业的试炼，我大致了解了完成一个较为复杂的程序的过程，积攒了宝贵的经验和教训。

**吴志承：**这次程设大作业我们组是制作一个游戏，因为没有做出多少贡献，所以可以说是抱了组长的大腿。我们组的游戏是“是蘑菇就下 100 层”，由于使用的是 Tkinter 库，画面效果不是很好。由这个大作业我最大的感想便是，有时候自己的想法用程序设计实现后效果完全和原来的不一样，另外，程序的优化非常的重要，并且会受到程序设计思想的限制。希望以后能做得更好。

