**广东海洋大学学生实验报告书（学生用表）**

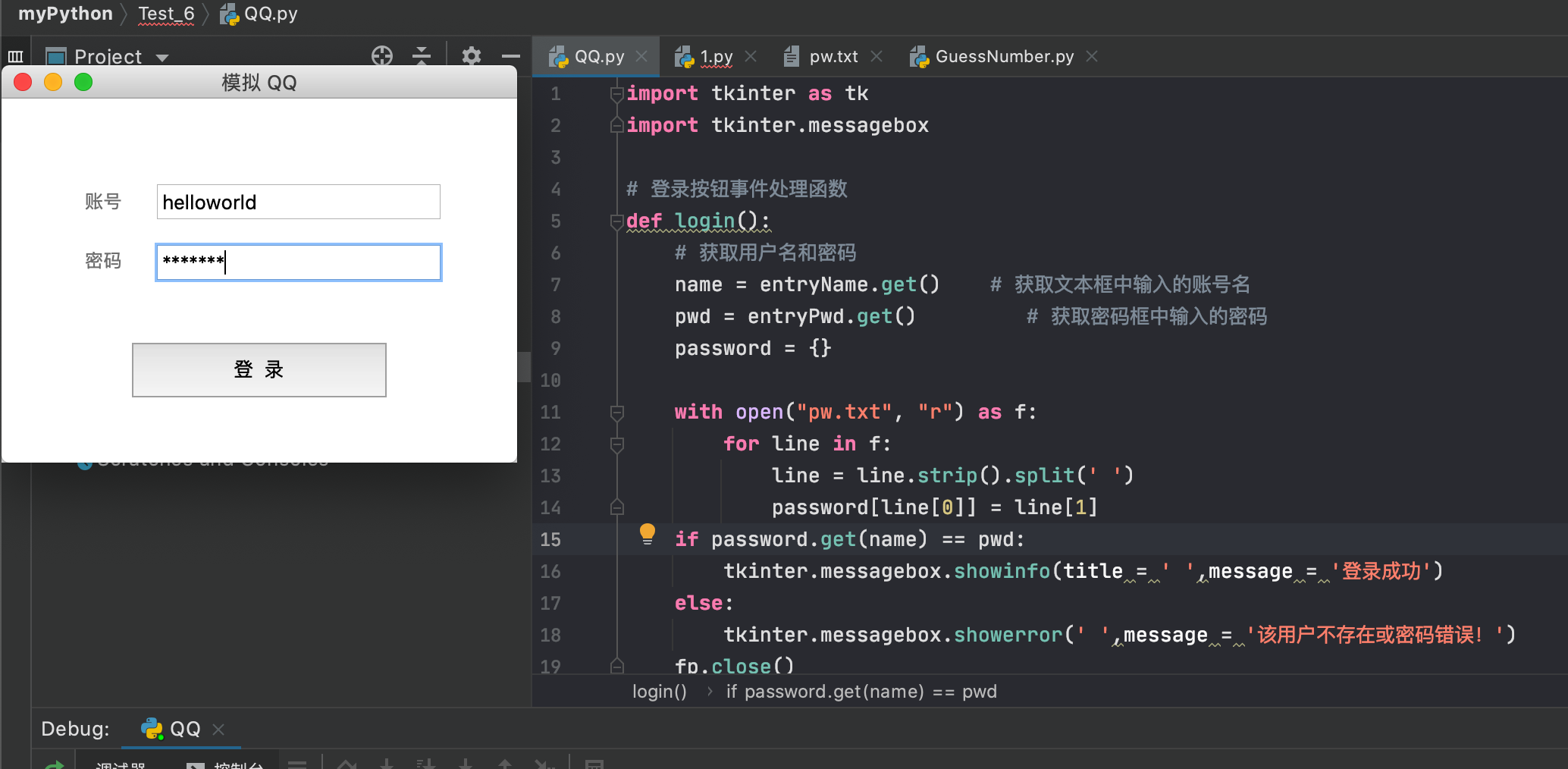
**GDOU-B-11-112**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | Python GUI编程 | | | | 课程名称 | | 高级程序设计语言 | | | | 课程号 | |  |
| 学院(系) | 数学与计算机学院 | | 专业 | 物联网工程 | | | | | 班级 | | 物联1181 | | |
| 学生姓名 | 陈用林 | 学号 | 201811672102 | | | 实验地点 | | 二教A501 | | 实验日期 | | 2020.11.24 | |

1. **实验内容：**

**1、设计一个窗体，并模拟QQ登录界面。将若干用户名和密码存储于文本文件（每行一条用户信息，用户名和密码用空格隔开）。登录时，若输入的用户名和密码存在于文件中，则显示“登录成功”消息；否则，提示“该用户不存在或密码错误！” 消息。**

**（1）程序代码：**



**import** tkinter **as** tk

**import** tkinter.messagebox

# 登录按钮事件处理函数

**def** login():

# 获取用户名和密码

name = entryName.get() # 获取文本框中输入的账号名

pwd = entryPwd.get() # 获取密码框中输入的密码

password = {}

**with** open("pw.txt", "r") **as** f:

**for** line **in** f:

line = line.strip().split(' ')

password[line[0]] = line[1]

**if** password.get(name) == pwd:

tkinter.messagebox.showinfo(title = ' ',message = '登录成功')

**else**:

tkinter.messagebox.showerror(' ',message = '该用户不存在或密码错误！')

fp.close()

# if name == 'helloworld' and pwd == '123456':

# tk.messagebox.showinfo(title='提示',

# message='登录成功')

# else:

# tk.messagebox.showerror('提示', message='该用户不存在或密码错误！')

# varName.set('') # 清空文本框

# varPwd.set('') # 清空密码框

# 第一步：创建应用程序窗口

window = tk.Tk()

window.title('模拟 QQ')

window.geometry('340x240') # 根据样图宽高比例，设定合适的窗口大小

# 第三步：创建账号密码的标签

labelName = tk.Label(window,

text='账号 ',

font=('微软雅黑', 12),

fg='grey',

justify=tk.RIGHT

)

labelName.place(x=50,y=57)

labelPwd = tk.Label(window,

text='密码 ',

font=('微软雅黑', 12),

fg='grey',

justify=tk.RIGHT

)

labelPwd.place(x=50,y=96)

# 第四步：创建字符串变量和文本框组件，同时设置关联的变量

varName = tk.StringVar(window, value='')

entryName = tk.Entry(window,

textvariable=varName)

entryName.place(x=100,y=54)

varPwd = tk.StringVar(window, value='')

entryPwd = tk.Entry(window,

show='\*',

textvariable=varPwd)

entryPwd.place(x=100,y=94)

# 第五步：创建按钮组件，同时设置按钮事件处理函数

buttonOk = tk.Button(window,

text='登 录',

relief='flat',

command=login)

buttonOk.place(x=85,y=160, width=170, height=38)

# 启动消息循环

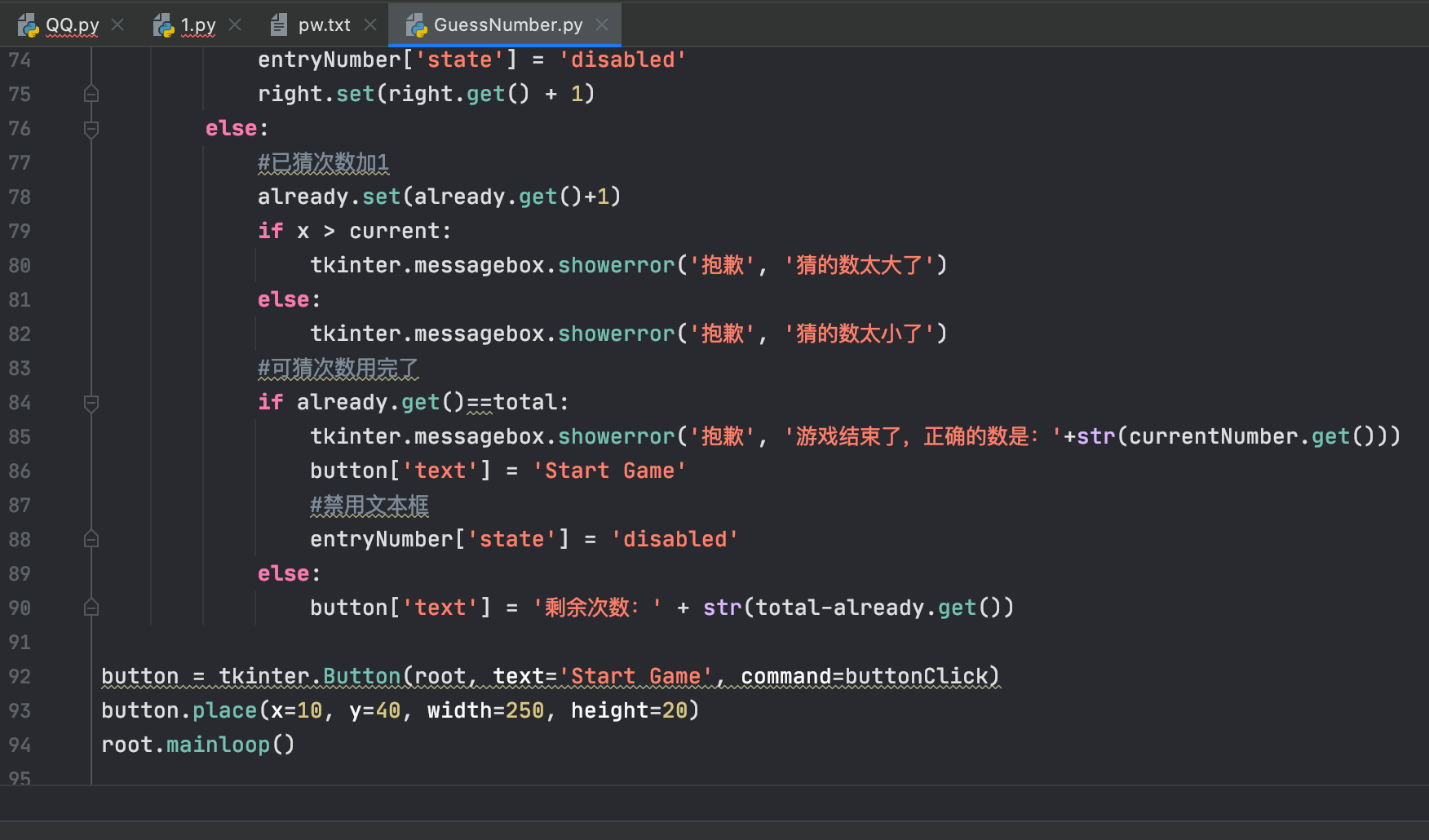
window.mainloop()

**（2）运行结果（截图）：**

|  |  |
| --- | --- |
|  |  |
|  |  |

**2、编写GUI版猜数游戏。启动游戏，由用户设置数值范围和猜测次数；退出游戏时显示战绩。下 图是参考的用户界面。**

**（1）程序代码：**



**import** random

**import** tkinter

**from** tkinter.messagebox **import** showerror, showinfo

**from** tkinter.simpledialog **import** askinteger

root = tkinter.Tk()

root.title('猜数游戏')

root .geometry('280x80+400+300')

root.resizable(**False**,**False**)

varNumber = tkinter.StringVar(root, value = '0')

totalTimes = tkinter.IntVar(root, value = 0)

already = tkinter.IntVar(root, value = 0)

currentNumber = tkinter.IntVar(root, value = 0)

times = tkinter.IntVar(root, value = 0)

right = tkinter.IntVar(root, value = 0)

lb = tkinter.Label(root, text = '请输入一个整数：')

lb.place(x = 10, y = 10, width = 100, height = 20)

entryNumber = tkinter.Entry(root, width=140, textvariable=varNumber)

entryNumber.place(x=110, y=10, width=140, height=20)

entryNumber['state'] = 'disabled'

**def** closeWindow():

message = '本次共玩游戏 {0} 次，猜对 {1} 次！\n欢迎下次再玩！'.format(times.get(), right.get())

tkinter.messagebox.showinfo('战绩', message)

root.destroy()

root.protocol('WM\_DELETE\_WINDOW', closeWindow)

**def** buttonClick():

**if** button['text']=='Start Game':

**while True**:

**try**:

start = tkinter.simpledialog.askinteger('允许的最小整数', '最小数', initialvalue=1)

**break**

**except**:

**pass**

**while True**:

**try**:

end = tkinter.simpledialog.askinteger('允许的最大整数', '最大数', initialvalue=10)

**break**

**except**:

**pass**

currentNumber.set(random.randint(start, end))

**while True**:

**try**:

t = tkinter.simpledialog.askinteger('最多允许猜几次？', '总次数', initialvalue=3)

totalTimes.set(t)

**break**

**except**:

**pass**

#已猜次数初始化为0

already.set(0)

button['text'] = '剩余次数：' + str(t)

#把文本框初始化为0

varNumber.set('0')

#允许用户开始输入整数

entryNumber['state'] = 'normal'

#玩游戏的次数加1

times.set(times.get() + 1)

**else**:

#一共允许猜几次

total = totalTimes.get()

#本次游戏的正确答案

current = currentNumber.get()

#玩家本次猜的数

**try**:

x = int(varNumber.get())

**except**:

tkinter.messagebox.showerror('抱歉', '必须输入整数')

**return**

**if** x == current:

tkinter.messagebox.showinfo('恭喜', '猜对啦')

button['text'] = 'Start Game'

#禁用文本框

entryNumber['state'] = 'disabled'

right.set(right.get() + 1)

**else**:

#已猜次数加1

already.set(already.get()+1)

**if** x > current:

tkinter.messagebox.showerror('抱歉', '猜的数太大了')

**else**:

tkinter.messagebox.showerror('抱歉', '猜的数太小了')

#可猜次数用完了

**if** already.get()==total:

tkinter.messagebox.showerror('抱歉', '游戏结束了，正确的数是：'+str(currentNumber.get()))

button['text'] = 'Start Game'

#禁用文本框

entryNumber['state'] = 'disabled'

**else**:

button['text'] = '剩余次数：' + str(total-already.get())

button = tkinter.Button(root, text='Start Game', command=buttonClick)

button.place(x=10, y=40, width=250, height=20)

root.mainloop()

**（2）运行结果（截图）：**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  | |

1. **实验总结：**

**（本次实验的知识点小结；实验体会；未解决的问题；改进的方法）**

通过本次试验，基本掌握了Python的GUI编程的方法、技巧和内容，理解了Tkinter标准库的用法，熟悉了Tkinter创建窗口和组件的方法、常用组件的属性及其设置方法、为窗体组建绑定鼠标事件的方法、使用Tkinter各类对话框等。

中间也遇到了问题，但是最终通过纠错还是成功完成对应的内容，代码都能正常运行，要求效果都能实现，获益良多。