《Python语言程序设计》实验报告

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 马昆 | 班级 | 2206831 | | | 学号 | 2206831544 |
| 实验名称 | 实验七 **Python图形开发库** | | | | | | |
| 实验时间 | 2023年 05 月 26 日 | | | 成绩 |  | | |
| **实验目的：**  1、熟悉tkinter编程基础  2、熟悉tkinter组件  3、掌握tkinter对话框  **实验环境：**  1、硬件环境：联想拯救者Y7000P-2022。  2、软件环境：  1)软件：Anaconda,spyder  2)操作系统：Windows 11 专业工作站版  **实验内容：**  编程练习：  1、使用tkinter开发猜数字游戏。游戏中计算机随机生成1024以内数字，玩家去猜，如果猜得数字过大过小都会提示，程序要统计玩家猜的次数。  2、扩展练习：游戏初步——扑克牌发牌程序窗体图形版。4名牌手打牌，计算机随机将52张牌（不含大小鬼）发给4名牌手，在屏幕上显示每位牌手的牌。  **实验过程及结果记录：**   1. 编程练习：   使用tkinter开发猜数字游戏。游戏中计算机随机生成1024以内数字，玩家去猜，如果猜得数字过大过小都会提示，程序要统计玩家猜的次数。   |  | | --- | | 代码  import tkinter as tk  import sys  import random  import re  number = random.randint(0,1024)  running = True  num = 0  nmaxn = 1024  nminn = 0  def eBtnClose(event):  root.destroy()  def eBtnGuess(event):  global nmaxn  global nminn  global num  global running  if running:  val\_a = int(entry\_a.get())  if val\_a == number:  labelqval("恭喜答对了！")  num+=1  running = False  numGuess()  elif val\_a < number:  if val\_a > nminn:  nminn = val\_a  num+=1  label\_tip\_min.config(label\_tip\_min,text=nminn)  labelqval("小了哦")  else:  if val\_a < nmaxn:  nmaxn = val\_a  num+=1  label\_tip\_max.config(label\_tip\_max,text=nmaxn)  labelqval("大了哦")  else:  labelqval('你已经答对啦...')  def numGuess():  if num == 1:  labelqval('我靠！一次答对！')  elif num < 10:  labelqval('= =十次以内就答对了牛。。。尝试次数：'+str(num))  elif num < 50:  labelqval('还行哦尝试次数：'+str(num))  else:  labelqval('好吧。。。。。您都试了超过50次了。。。。尝试次数：'+str(num))  def labelqval(vText):  label\_val\_q.config(label\_val\_q,text=vText)  root = tk.Tk(className="猜数字游戏")  root.geometry("400x90+200+200")  line\_a\_tip = tk.Frame(root)  label\_tip\_max = tk.Label(line\_a\_tip,text=nmaxn)  label\_tip\_min = tk.Label(line\_a\_tip,text=nminn)  label\_tip\_max.pack(side = "top",fill = "x")  label\_tip\_min.pack(side = "bottom",fill = "x")  line\_a\_tip.pack(side = "left",fill = "y")  line\_question = tk.Frame(root)  label\_val\_q = tk.Label(line\_question,width="80")  label\_val\_q.pack(side = "left")  line\_question.pack(side = "top",fill = "x")  line\_input = tk.Frame(root)  entry\_a = tk.Entry(line\_input,width="40")  btnGuess = tk.Button(line\_input,text="猜")  entry\_a.pack(side = "left")  entry\_a.bind('<Return>',eBtnGuess)  btnGuess.bind('<Button-1>',eBtnGuess)  btnGuess.pack(side = "left")  line\_input.pack(side = "top",fill = "x")  line\_btn = tk.Frame(root)  btnClose = tk.Button(line\_btn,text="关闭")  btnClose.bind('<Button-1>',eBtnClose)  btnClose.pack(side="left")  line\_btn.pack(side = "top")  labelqval("请输入0到1024之间任意整数：")  entry\_a.focus\_set()  print(number)  root.mainloop() | | 运行结果1    运行结果2 |  1. 扩展练习   游戏初步——扑克牌发牌程序窗体图形版。4名牌手打牌，计算机随机将52张牌（不含大小鬼）发给4名牌手，在屏幕上显示每位牌手的牌   |  | | --- | | 代码  import tkinter as tk  from tkinter import messagebox  import random  # 扑克牌花色和点数  suits = ['黑桃', '红桃', '方片', '梅花']  ranks = ['A', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'J', 'Q', 'K']  cards = [f"{suit}{rank}" for suit in suits for rank in ranks]  # 洗牌并发牌函数  def deal\_cards():  random.shuffle(cards)  player\_hands = [[] for \_ in range(4)]  for i, card in enumerate(cards):  player\_hands[i % 4].append(card)    # 显示每位牌手的牌  for i, hand in enumerate(player\_hands):  player\_cards = ' '.join(hand)  player\_label = tk.Label(root, text=f"玩家 {i+1}: {player\_cards}")  player\_label.pack()  # 创建窗体  root = tk.Tk()  root.title("发牌模拟")  root.geometry("600x300")  # 创建发牌按钮  deal\_button = tk.Button(root, text="点击发牌", command=deal\_cards)  deal\_button.pack(pady=10)  root.mainloop() | | 运行结果1      运行结果2 | | | | | | | | |
| **实验总结：**  通过这个实验，学习了如何使用Tkinter库创建简单的GUI程序。了解到GUI编程的基本概念，包括创建窗体、添加按钮和标签等元素，并将它们与事件绑定。并且还学习了使用布局管理器进行界面布局。本次实验也巩固了之前学到的一些基础知识，例如列表、循环和函数。 | | | | | | | |
|  | | | | | | | |