Int-213 Python Programming

CA-2 project

Project title: rolling dice
Simulator

Submitted to: Sagar pande sir

Submitted by: Nellam Sai Teja,
Kusumaraju Vamsi

Contents page:

Introduction:

Firstly, we have selected rolling dice simulator as my project. We made this project using Tkinter. In this project we can roll multiple dices as per our choice but less than or equal to 15 dices. We had divided the project into parts . in the first part we have done the user interface after implementing and testing parallaly we had written code for rolling the dice and then we made it as a single project.

Project Description:

When we run the project we will be seeing an user interface. The user interface shows us dices and a heading of "rolling dice simulator" and a button showing "click here to go". When we click that button we will be redirecting to another page, in that page it shows "how many dice you want to roll(from 1 to 15)" with an entry box next to it. The entry box is by default set to "0". The page also displays three buttons. On the first button it labels as "click to roll" after entering the number of dices in the entry box and then clicking on the button "click to roll" it shows the same number of dices that we provided in the entry box rolled on the screen randomnly.

When we want to change the dices we have to click on the button labeled "reselect".after clicking on it we will be getting an fresh page as mentioned above containing headings and buttons with a default dices as zero. And then we can enter our required number of dices. When we enter dices more than 15 or less than 1 then it displays a warning to select valid number of dices. We also providea a button labeled "exit" when we click on this button it asks us , "are you sure to exit?" displaying yes and no. when we click on yes it will exit,when we click on no we will be in the same page.

Explaination of code:

Code for landing page: this is the code for the page that displays after running the programme.

```
def landing_page(self):


    canvas = Canvas(self.win, width=self.width, height=self.height)

    image = ImageTk.PhotoImage(Image.open("background.jpg"))

    canvas.create_image(0, 0, anchor="nw", image=image)

    canvas.place(relx=0, rely=0)

    welcome_label = Label(self.win, text=" Rolling Dice Simulator ", fg="white", bg="black",

                          font=('Helvetica', '50'))

    welcome_label.place(relx='0.28', rely='0.1')

    start = "Click here to Go."

    start_button = Button(self.win, text=start, fg="white", cursor='hand2', bg="black", font=('Helvetica', '30'),

                          command=lambda: self.simulate_page())

    start_button.place(relx="0.4", rely='0.8')

    self.win.mainloop()
```

code for the simulation page: this code is for the page that redirects when we click on “click here to go”

```
def simulate_page(self):


    global simulate_window  # Declaring globally another window
    simulate_window = Toplevel(self.win)  # Creating another window above first
window
    simulate_window.title("Simulation_page")  # Title for the window
    # Window background
    simulate_window.config(bg="salmon4")
    # Window geometry
    width = simulate_window.winfo_screenwidth()
    height = simulate_window.winfo_screenheight()
    simulate_window.geometry(f'{width}x{height}')
    dot = Label(simulate_window, text=" ", fg="black", bg="salmon4",
                font=('Helvetica', '25'))
    dot.grid(row=0, column=0)
    ques = "How many dice you want to Roll(from 1 to 15)"  # Prompt text
    question_label = Label(simulate_window, text=ques, fg="white",
bg="salmon4",
                           font=('Helvetica', '25'))  # Label to ask number of dies
    question_label.place(relx=0.01, rely=0.01)  # Placing label at a particular place
    global answer_entry
    answer_entry = IntVar()
    answer_entry.set(0)  # Setting default value for entrybox
```

```
        answer_entry = Entry(simulate_window, textvariable=answer_entry, width=2,
fg="black", bg="white",

                             font=('Helvetica', '25'))  # Entry box for taking input

        answer_entry.place(relx=0.44, rely=0.01)  # Placing  entry box



        Text = "Click to Roll."
```

Code for rolling die: the following code is written for rolling the dice.

```
def roll_die():  # function for rolling a die

            no_of_dice = int(answer_entry.get())  # Storing  answer into refrence
variable

            try:

                if no_of_dice < 1 or no_of_dice > 15:

                    messagebox.showinfo("Warning", "Please select valid number of Dies!")

                    return



                print(no_of_dice)

                dice = ['dice_1.jpg', 'dice_2.jpg', 'dice_3.jpg', 'dice_4.jpg', 'dice_5.jpg',
'dice_6.jpg']

                try:

                    for i in range(1, no_of_dice + 1):

                        die = random.choice([0, 1, 2, 3, 4, 5])

                        print(die)

                        canvas = Canvas(simulate_window, width=240, height=240)

                        image = ImageTk.PhotoImage(Image.open(dice[die]))

                        canvas.create_image(0, 0, anchor="nw", image=image)
                        canvas.image = image
                        if i > 10:
                            canvas.grid(row=3, column=i - 11, pady=10, padx=10)
                        elif i > 5:
                            canvas.grid(row=2, column=i - 6, padx=10, pady=10)
                        else:
                            canvas.grid(row=1, column=i - 1, padx=10, pady=10)
                except ValueError as v:
                    messagebox.showinfo("Warning", "Please select valid number of Dies!")


            except ValueError as v:
                messagebox.showinfo("Warning", "Please select valid number of Dies!")


        roll_button = Button(simulate_window, text=Text, fg="black", bg="white",
                             font=('Helvetica', '15'), command=lambda: roll_die(),
cursor='hand2')
        roll_button.place(relx=0.6, rely=0.005)
        reroll_button = Button(simulate_window, text="Reselect", fg="black",
bg="white",
                               font=('Helvetica', '15'), command=lambda: self.simulate_page(),
cursor='hand2')


        exit_button = Button(simulate_window, text="Exit", fg="white", bg="red",
activebackground='red',
                             font=('Helvetica', '20'), command=lambda: exit(), cursor='hand2')
        reroll_button.place(relx=0.7, rely=0.005)
```

```
    exit_button.place(relx=0.78, rely=0.005)


    simulate_window.mainloop()
```

code for exit function: this code is written to exit the page.

```
def exit():

      response = messagebox.askyesno("Done", "Are you sure to Exit.?")  # Message for  asking yes or no

      print(response)

      if response:

         simulate_window.destroy()  # It destroys window

         self.win.destroy()
```

conclusion:

to conclude we not only feel this as an assignment, but an opportunity for us to develop ourselves further within the field of computer programming. We were faced with challenges at each step of the way and were required to push ourselves further each time. After finishing the parts we made it errorfree and we made it as a project.