MINOR PROJECT REPORT
ON
DICE ROLLING SIMULATOR

BACHELOR OF ENGINEERING

In

Computer Science and Engineering

SUBMITTED BY:

Group no.- 1(G6)

Riya Gupta (2020A1R181)

Anchal Baghat (2020A1R187)

Swasti Gupta (2020A1R191)

Shrifal Raina (2020A1R177)

Model Institute of Engineering and Technology

Kot Bhalwal, Jammu - 181122

Contents

What is a Dice Rolling Simulator?

The Dice is a simple cube shaped like object used in various board games such as snake-ladder, Ludo etc. It is a simple cube which generates a random number when rolled by a user. A Dice rolling simulator is nothing but a computer model that can be created by a software program and it functions the same as a normal dice in which the user rolls a dice and a random number gets shown on the screen. There are many ways a dice simulator can be implemented. In this paper, we will implement a GUI Application of dice simulator using Tkinter Framework in Python.

What is Tkinter?

Python has various frameworks/packages which can be used for making GUI applications i.e., the Graphical User Interface. Tkinter is the standard GUI package for Python. It is the most common, fast, and easy way to create a GUI (Graphical User Interface) application. It provides a powerful object-oriented interface to the Tk GUI toolkit. We can develop an application and can use that application on any platform, which reduces the need for additional requirements required to use an application on any OS like Windows, Mac, or Linux.

What is GUI (Graphical User Interface)?

The Graphical User Interface (GUI) is a form of user interface which allows a user to interact with electronic devices with the help of graphical icons and visual components. It displays objects that conveys information, and displays actions performed by the user.

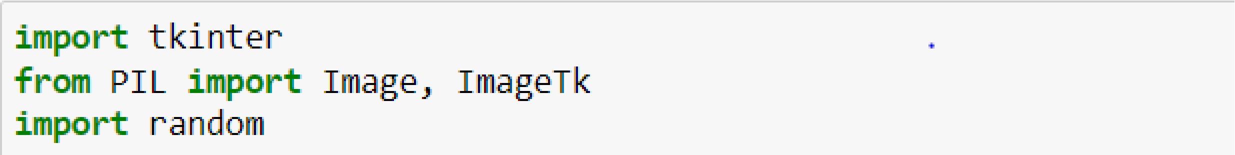
Dice Rolling Simulator Project in Python with Source Code

- **Step 1: Importing the required module**

  We will import the following modules:

  - Tkinter: Imported to use Tkinter and make GUI applications.
  - Image, Imagetk: Imported from PIL, i.e. Python Imaging Library. We use it to perform operations involving images in our UI.
  - Random: Imported to generate random numbers.

Code:

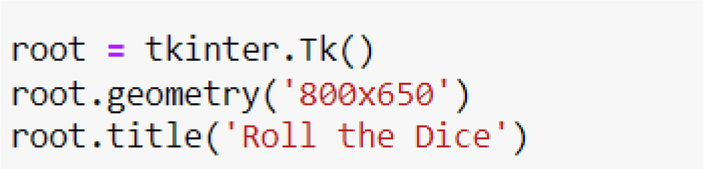


```
import tkinter
from PIL import Image, ImageTk
import random
```

- **Step 2: Building a top-level widget to make the main window for our application**

In this step, we will build the main window of our application, where the buttons, labels, and images will reside. We also give it a title by title() function.

Code:



```
root = tkinter.Tk()
root.geometry('800x650')
root.title('Roll the Dice')
```

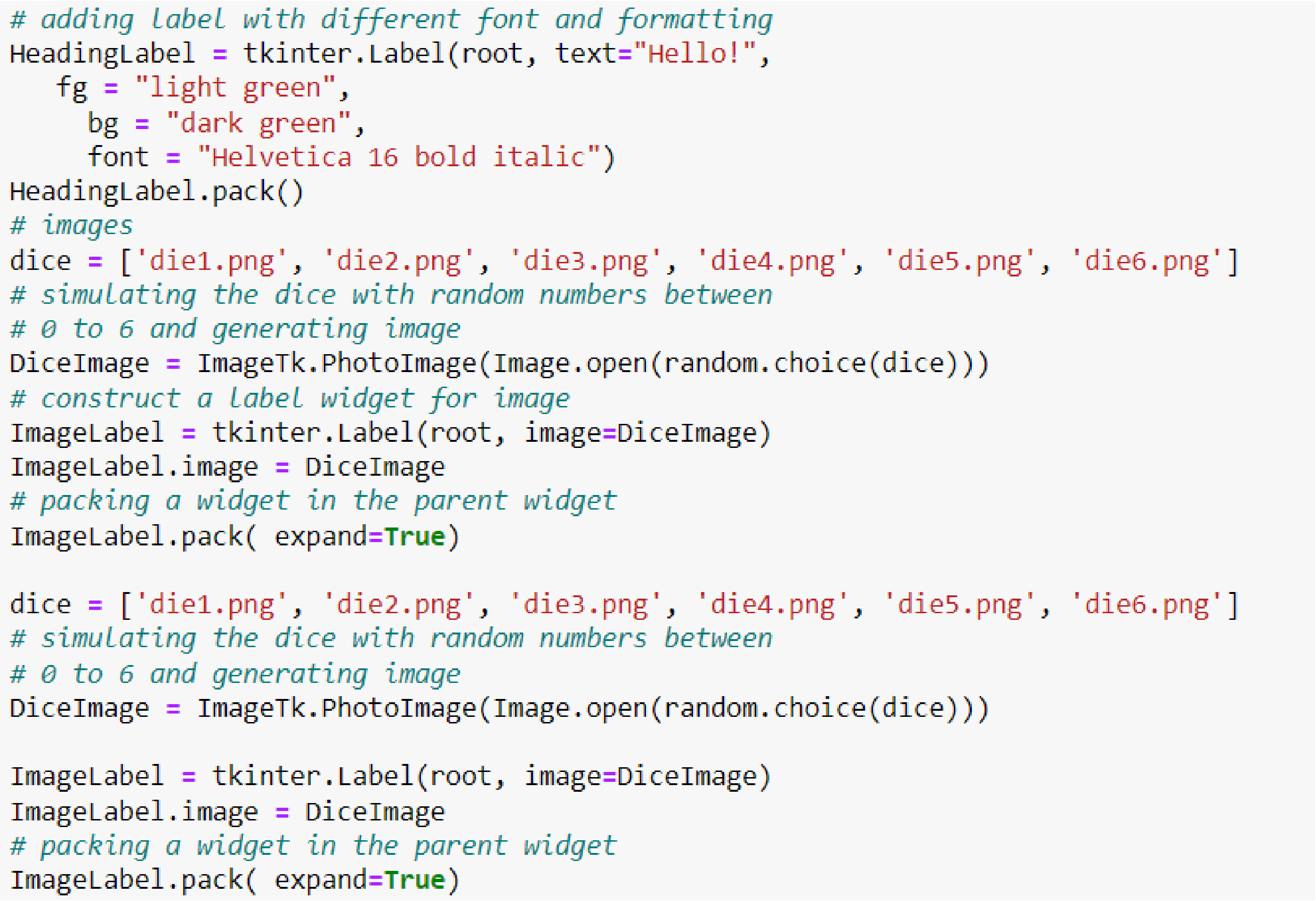
Explanation:

The above code sets the title of the application window as ‘ Roll the Dice’. Running the above code will generate a blank window of a dice rolling simulator python project with the title on it.

- **Step 3: Designing the buttons**

The below code will add a label giving a heading to our dice simulator. Also, we will add an image area, which will display the image chosen by random numbers.

Code:



```
# adding label with different font and formatting
HeadingLabel = tkinter.Label(root, text="Hello!",
   fg = "light green",
     bg = "dark green",
     font = "Helvetica 16 bold italic")
HeadingLabel.pack()
# images
dice = ['die1.png', 'die2.png', 'die3.png', 'die4.png', 'die5.png', 'die6.png']
# simulating the dice with random numbers between
# 0 to 6 and generating image
DiceImage = ImageTk.PhotoImage(Image.open(random.choice(dice)))
# construct a label widget for image
ImageLabel = tkinter.Label(root, image=DiceImage)
ImageLabel.image = DiceImage
# packing a widget in the parent widget
ImageLabel.pack( expand=True)

dice = ['die1.png', 'die2.png', 'die3.png', 'die4.png', 'die5.png', 'die6.png']
# simulating the dice with random numbers between
# 0 to 6 and generating image
DiceImage = ImageTk.PhotoImage(Image.open(random.choice(dice)))

ImageLabel = tkinter.Label(root, image=DiceImage)
ImageLabel.image = DiceImage
# packing a widget in the parent widget
ImageLabel.pack( expand=True)
```

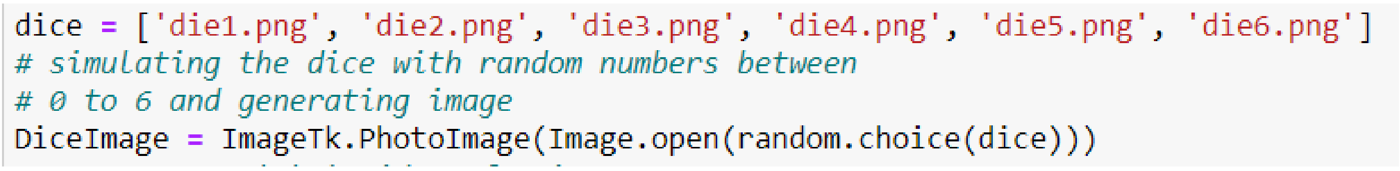
Explanation:

Here, we use pack() to arrange our widgets in row and column form. The ‘BlankLine’ label is to skip a line, whereas we use the ‘HeadingLabel’ label to give a heading.

- root – the name by which we refer to the main window of the application
- text – text to be displayed in the HeadingLabel
- fg– the colour of the font used in HeadingLabel
- bg – background colour of the HeadingLabel
- font – used to give customised fonts to the HeadingLabel text
- .pack() – Used to pack the widget onto the root window

- **Step 4: Forming a list of images to be randomly displayed**

Code:



```
dice = ['die1.png', 'die2.png', 'die3.png', 'die4.png', 'die5.png', 'die6.png']
# simulating the dice with random numbers between
# 0 to 6 and generating image
DiceImage = ImageTk.PhotoImage(Image.open(random.choice(dice)))
```

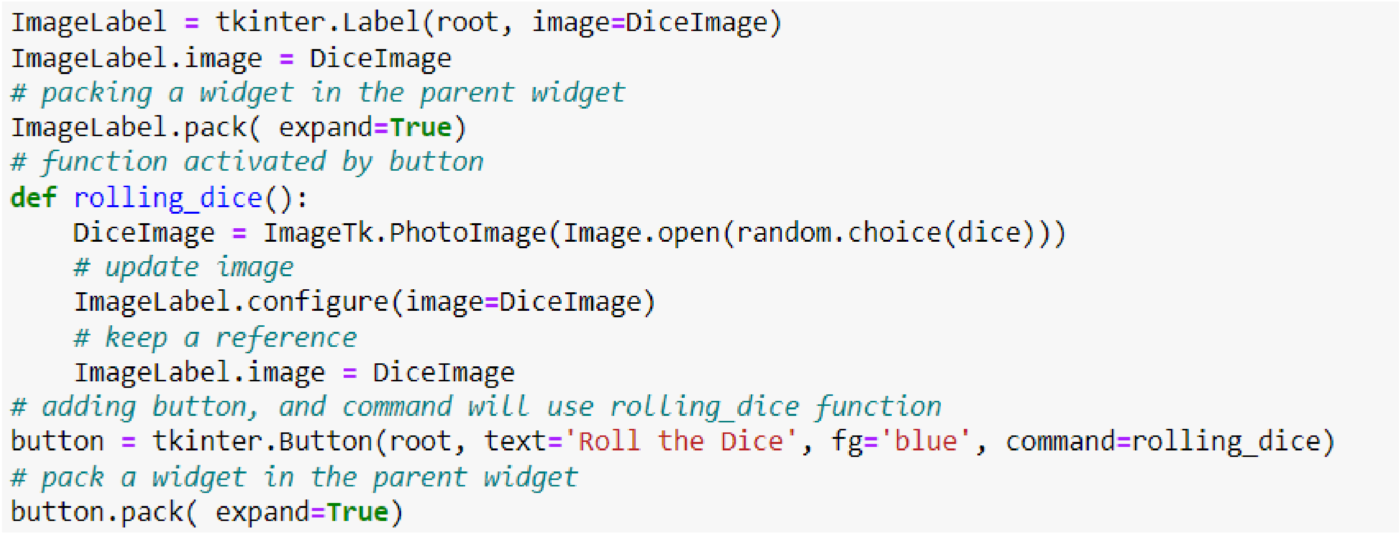
Explanation:

‘dice’ is the list of names of images kept in the same folder, which are chosen randomly according to the random number generated.

‘DiceImage’ is used to store an image of dice which is chosen by randomly generated numbers.

- **Step 4: Constructing a label for image, adding a button and assigning functionality**

Code:



```
ImageLabel = tkinter.Label(root, image=DiceImage)
ImageLabel.image = DiceImage
# packing a widget in the parent widget
ImageLabel.pack( expand=True)
# function activated by button
def rolling_dice():
    DiceImage = ImageTk.PhotoImage(Image.open(random.choice(dice)))
    # update image
    ImageLabel.configure(image=DiceImage)
    # keep a reference
    ImageLabel.image = DiceImage
# adding button, and command will use rolling_dice function
button = tkinter.Button(root, text='Roll the Dice', fg='blue', command=rolling_dice)
# pack a widget in the parent widget
button.pack( expand=True)
```

Explanation:

‘ImageLabel’ is to place an image in the window. The parameter expands declared as True so that even if we resize the window, the image remains in the center.

Major function:

‘rolling_dice’ function is a function that is executed every time a button is clicked. This is attained through the ‘command=rolling_dice’ parameter while defining a button.

- **Step 6: Forming a list of images to be randomly displayed**

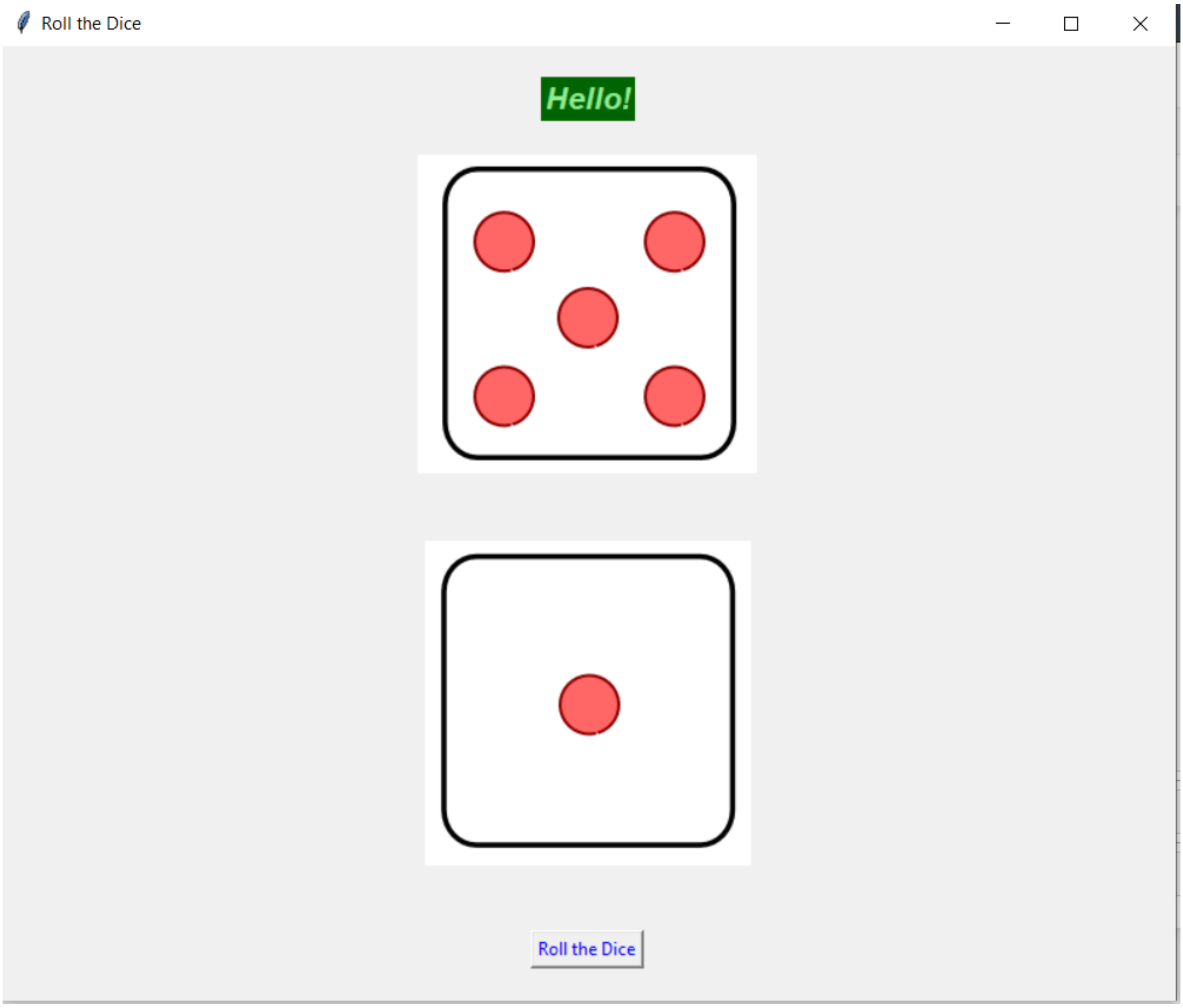
Code:

```
# call the mainloop of Tk
# keeps window open
root.mainloop()
```

Explanation:

‘root.mainloop()’ is used to open the main window. It acts as the main function of our program.

Dice Rolling Simulator Python Output



References:

- https://en.wikipedia.org/wiki/Tkinter
- https://www.tutorialspoint.com/python/python_gui_programming.htm
- https://www.w3schools.com/python/ref_random_choice.asp
- https://data-flair.training/blogs/dice-rolling-simulator-python/
- https://pillow.readthedocs.io/en/stable/reference/ImageTk.html
- http://clipart-library.com/dice-faces.html