HANDS-ON PYTHON(CSE-106L)

Dept. of Computer Science and Engineering

III Semester

PROJECT REPORT

MINI PROJECT USING PYTHON

ON

DIGITAL CLOCK

Abstract

Digital Clock is a **GUI** application, which displays the actual time in a pop window (i.e., Widget) and it is a designed application where there is a background color and foreground color and which we can use this application in daily life. For example, it can be used in railway stations and airports and big displays in Metropolitan cities. The code which we have is used to display the real-time time purpose only.

Our main aim is to make the digital clock in the simplest, accurate, and creative manner by using a simple code that can be processed using low-configured machines by the user approach.

System Requirement

The minimum requirements which are used to run a python application are suitable to run the program. This code can be processed using raspberry pi. **System Architecture**

- Modern Operating System:
- Windows 7 or 10
- Mac OS X 10.11 or higher, 64-bit
- Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)
- x86 64-bit CPU (Intel / AMD architecture)
- 4 GB RAM
- 5 GB free disk space

Most users will find that any computer bought in recent years will meet (and usually exceed) these hardware requirements.

Algorithm

- We have imported some of the libraries to build the application
- We have imported the libraries to design the window
- We have used the Tkinter function and we have designed the label
- Here the coolest work is to design the label where we can put our own design in the window. This makes our work different from others.
- We have made the window size and made a
  Using font, a font which looks easy way to read the time
- The Border width of the text and the border of the window is arranged in a manner that does not look as weird.
- We made a window and designed it with the background color black and the time is displayed in sky blue color which looks more attractive in a creative manner.
- By doing all the above mentioned we have made our project in the most efficient approach.

Import Libraries

We will use two libraries in this project. And both of them come with Python, which means we don't have to install them. These kinds of libraries are called Python built-in packages.

The main package we will use is Tkinter.

So for this step, all we need to do is to import them to our program:

```
from tkinter import Label, Tk
import time
```

Designing the Application Window

In this step, we will first define the window panel using Tkinter package. And after that, we will define the text design that we want to use for the digital clock.

Define the Window

As mentioned earlier, we will use Tkinter package. Tkinter is can be defined as Tk. And after defining it, we will customize it.

```
app_window = Tk()
app_window.title("My Digital Time")
app_window.geometry("420x150")
app_window.resizable(1,1)
```

Understanding the code:

- Defining the Tkinter function.
- Giving a title to our application window.
- Defining the size of our video, for example in my case, it's 420pixels width to 150pixels height.
- The window is not resizable, because the text values are not responsive in design. And we don't want our design to look weird when the window size is changed.

Perfect, our application window is ready! Now, let's work on the clock design.

The Label Design

The cool step of the program is this one. Because you can put your own preferences into the design. This step will make your work different from others. If you love designing things, it's time to show off your skills.

There are four elements that we will customize:

- The font of the digital numbers.
- The background colour of our digital clock.

- The colour of the digital numbers, make sure it is not the same colour as your background.
- The border width of the text.

Here are the values that I used for my design:

```
text_font= ("Boulder", 68,
'bold') background = "#f2e750"
foreground= "#363529"
border_width = 25
```

For colors, feel free to use RGB values or hex values. In my case, I used the hex values of the colors. I use google's color picker that is available on the browser. Just search "Color picker" on google search. And you will see it.

Now, let's combine the elements and define our label. Label function is the text that will show our time.

```
label = Label(app_window, font=text_font, bg=background,
fg=foreground, bd=border_width) label.grid(row=0, column=1)
```

Digital Clock Function

If we are working on an application project, functions are the best way to make things work. Functions are also great because they make the program more structured and easier to understand.

Alright, let's define our digital clock function then:

```
def digital_clock():
   time_live = time.strftime("%H:%M:%S")
label.config(text=time_live)
label.after(200, digital_clock)
```

Understanding the code:

- In the first line, we are getting real-time using the time package. And we are also defining the format that we want it to be. Since we are designing a digital clock, “hour, minutes, seconds” will be a nice format to go with.
- In the second line, we are just assigning the real-time to the label method. This way the digital time will be updated.
- And lastly, we are calling the function again so that the digital clock is showing the live time. This way every 200 milliseconds the time is getting updated. In programming, this is called a recursion loop. Calling the same function, inside the function. Feels like inceptions, isn’t that cool?

Run the Application

Great! You made it until this step, which is the final step of our application project. As you know functions will not run unless you call them. To trigger the application, we will call the function. Let’s run the application:

```
digital_clock()
app_window.mainloop()
```

Result

```
python digital_clock.py
```

Code:

```
#tkinter or toolkit interface is a library to create Graphic User Interface.
# Without it, also there have two modules, which is wxPython (Highly
functional user interface)
# and JPython (A java implementation of Python).
from tkinter import Label, Tk import time
app_window = Tk()
# Using title function we just define the name of the main
windows. app_window.title("Digital Clock")
app_window.geometry("420x150") app_window.resizable(1,1)
# label is a widget, where you place text, image and more attribute of
design. This function is divided into two part.
# One is master, represent the parent window. And another one is option, such
as font, background, foreground, height,
# weight etc. All of option are separated by comma. peck() function is a
geometry manager, which is the widget position.
# Here we use fill option and assign a X value. it's mean fill horizontally.
We could use Y (fill vertically) or both.
# Also there has a 'None' (Default) variable. Even 'expand' and 'side'
another two option, which I could use.
text_font= ("Boulder", 68, 'bold')
background = "#f2e750" foreground=
"#363529" border_width = 25


label = Label(app_window, font=text_font, bg=background, fg=foreground,
bd=border_width)
label.grid(row=0, column=1)
 def
digital_clock():
   # Let have a define a function, name is time. Here, strftime()
function return 4 parameter taken from local time.    time_live =
time.strftime("%H:%M:%S")    label.config(text=time_live)
label.after(200, digital_clock)


# Call the function, which I have created.
digital_clock()
# The mainloop() function is used to keep run the application, wait for an
event occur as long as the windows is close.
app_window.mainloop()
```

OUTPUT SCREEN

Conclusion:

This Project is mainly based on the GUI Application. In this project, we use the Tkinter module package and we used to import the time for running the application most accurately. This can also be used as a widget application in Windows or Mac. This application is mainly used in Railways and Airports and Public Transportation to display the current time.