**MINI PROJECT**

**OPEN SOURCE TECH LAB (PERL)**

**1.Sriram Venkatesh 66**

**2.Ankita Singh 60**

**3.Shubham Singh 63**

**4.Suraj Tiwari 65**



Department of Computer Engineering  
St. John College of Engineering and Management  
University of Mumbai

2017-2018

**INTRODUCTION**

Perl is one of the most popular languages out there, and is used for everything from mission-critical projects to Web applications to "glue." It is not, however, often used for **GUI programming** and prototyping. **Philipp K. Janert** thinks it should be, and you probably will too –after this look at some of the more complex widgets available for Perl/Tk.

In contrast to the widespread use of the Perl language, Perl's GUI toolset, Perl/Tk, is much less popular. This is strange, because it is arguably one of the easiest GUI toolkits to program (at least for **UNIX platforms**), and therefore suggests itself for user interface prototyping or for quickly providing a user-friendly wrapper around cryptic command-line tools.

One reason for this relative indifference toward Perl/Tk appears to be the perception that it is **not very powerful** and does not lend itself to sophisticated applications. However, a number of widgets that provide more complex functionality are available as user contributions on **CPAN**. In this article, I will take a look at some of them and show how they can be used to create a richer

user experience. I will also point out some more generally useful techniques for programming with Perl/Tk.

The other problem with Perl/Tk, of course is, that its widgets look **ugly** and are not **"theme-able."**

The role of **Tk** is to make the process of designing a user interface significantly easier. The core of the windowing system provides the methods and basis for simple operations and events, such as opening windows, drawing lines, and accepting input and actions from the keyboard and mouse.

Developing a user interface with Tk is a case of creating a number of nested objects. The first object you create is the main window for your application. The nested objects are the individual widgets that make up the user interface. A widget is a button, text box, menu, or any of a variety of other components used to build up your interface within your window.

**PROBLEM STATEMENT**

In this project, we will be creating a GUI based ADDRESS BOOK using Tkinter in Perl and also there will be two buttons: “ENTER” and “CLEAR”.

On Clicking Enter Button, the data will be stored/written in a file called as

“entry.txt”. Clear Button will clear all the Fields. After clicking both buttons a message will be pop-up on the screen.

**MODULES USED**

The **“Tkx”** module is used in this program. This module is not present by default in the basic Perl Installation. You can **install** these by performing

following steps:

1. Type the command **“ PS C:\Users\Suraj Tiwari>PPM**” in command prompt/Terminal/Windows PowerShell.

2. This will open a window containing all the modules for Perl.

3. From there, you can install the desired module(In this case, it is Tkx module).

**Tkx module** contains a main window which will run when the execution starts. This window contains all the widgets to be used for the user interaction.

**->The basic process for creating a Tk-based GUI application is as follows:**

**1.** Create a window to hold all of your objects.

**2.** Create a number of widgets, defining their contents, actions, and other

Elements.

**3.** Display and arrange the widgets within the window. This is handled by the

Pack geometry manager.

**4.** Start the event loop. The main execution of the script has now finished, and

The rest of the script will be driven by the events configured for individual

widgets.

**Some of the widgets used in this program are as follows:**

**1.ENTRY:** This widget will take the input same as like Textbox in HTML.It is a filed where a user can enter the data.

**Syntax:**

Tkx::ttk\_\_entry("widget\_name", options…);

**2.LABEL:** This widget will display the data to the screen. It is a non-editable text.

**Syntax:**

Tkx::ttk\_\_label("widget\_name", options)

**3.BUTTON:** This widget will display the button on the screen.

**Syntax:**

Tkx::ttk\_\_button(".c.clear", -text => "Clear", -command => sub {clear();});

-The **“text”** attribute will display the value on the Button.

-The **“command”** attribute will invoke the command (In this case,it is a subroutine) when the button will be clicked.

**4.MESSAGE\_BOX:** This widget will display a message box whenever it is called. There are various types of “messagebox” like: “yesno”,”okcancel”,etc.

**Syntax:**

Tkx::tk\_\_\_messageBox(-title=>"Conformation",-message => "Your Record has been Entered Successfully!!!");

-The **“title”** attribute will display the value as the title of the message box displayed.

-The **“message”** attribute will display the message on the pop-up.

**5.FRAME:** This widget will create a Frame in the window. A window can have Multiple Frames.

**Syntax:**

Tkx::ttk\_\_frame(".frame\_name", -padding => "30 30 30 30");

**-**We can Specify the name of the frame by writing it in double quotes preceded by a dot(.).

**-**The **“padding”** attribute will provide the co-oridnates for starting and ending of the Frame.

In this program we have also used a **“grid”** method.This will divide the window into number of grids.Thus now we can place the different widgets in the grids. This will align the widgets in a straight manner and thus we don’t need to provide the co-ordinates for the widgets placing.

**-Also we can specify font that is to be used while displaying the data.**

**Syntax:** Tkx::font\_create("font", -family => "font\_family", -size =>font\_size);

**IMPLEMENTATION**

**1.**Firstly we created a frame with name “.c” and padding is specified for the frame.

**2**.Later ,we created grids in the Frame. Also, we have specified a font : family=**Helvetica**, size=”12” , weight=”bold”.

**3.**We have created **5 labels** and their **5 corresponding Entry** to enter the value. The “row” and “column” value for the widgets are specified ,for fitting the widget in the grid ,to make a good-viewing and Interactive window.

**4**.There are two **buttons** specified . First one is **“Enter”** . On clicking this button, the data will be written in to the file and message is displayed that “**Your Record has been Entered Successfully!!!**”.

**5.**Second one is **“Clear”**. On clicking this button,the value of the Entry boxes will be reset to NULL values and message box is displayed “**Book Cleared Successfully!!!**”.

**Code**

use Tkx;

Tkx::wm\_title(".", "ADDRESS BOOK");

Tkx::ttk\_\_frame(".c", -padding => "30 30 30 30");

Tkx::grid( ".c", -column => 0, -row => 0, -sticky => "NWES");

Tkx::grid\_columnconfigure( ".", 0, -weight => 1);

Tkx::grid\_rowconfigure(".", 0, -weight => 1);

Tkx::font\_create("font", -family => "Helvetica", -size => 12, -weight => "bold");

Tkx::ttk\_\_entry(".c.name", -width => 25, -textvariable => \$name, -font => "font");

Tkx::grid(".c.name", -column => 2, -row => 1, -sticky => "W");

Tkx::ttk\_\_entry(".c.age", -width => 25, -textvariable => \$age, -font => "font");

Tkx::grid(".c.age", -column => 2, -row => 2, -sticky => "W");

Tkx::ttk\_\_entry(".c.address", -width => 25, -textvariable => \$address, -font => "font");

Tkx::grid(".c.address", -column => 2, -row => 3, -sticky => "W");

Tkx::ttk\_\_entry(".c.number", -width => 25, -textvariable => \$number, -font => "font");

Tkx::grid(".c.number", -column => 2, -row => 4, -sticky => "W");

Tkx::ttk\_\_entry(".c.email", -width => 25, -textvariable => \$email, -font => "font");

Tkx::grid(".c.email", -column => 2, -row => 5, -sticky => "W");

Tkx::ttk\_\_button(".c.enter", -text => "Enter", -command => sub {enter();});

Tkx::grid(".c.enter", -column => 1, -row => 6, -sticky => "W");

Tkx::ttk\_\_button(".c.clear", -text => "Clear", -command => sub {clear();});

Tkx::grid(".c.clear", -column => 2, -row => 6, -sticky => "WE");

Tkx::grid( Tkx::ttk\_\_label(".c.a", -text => "NAME: ", -font => "font"),-column => 1, -row => 1, -sticky => "W");

Tkx::grid( Tkx::ttk\_\_label(".c.b", -text => "AGE:", -font => "font"), -column => 1, -row => 2, -sticky => "W");

Tkx::grid( Tkx::ttk\_\_label(".c.d", -text => "ADDRESS : ", -font => "font"),-column => 1,-row => 3, -sticky => "W");

Tkx::grid( Tkx::ttk\_\_label(".c.e", -text => "CONTACT NO. : ", -font => "font"), -column => 1, -row => 4, -sticky => "W");

Tkx::grid( Tkx::ttk\_\_label(".c.f", -text => "E-MAIL ID:",-font => "font"), -column => 1, -row => 5, -sticky => "W");

sub enter{

$filename = 'entry.txt';

open($WRITEFILE, '>>', $filename);

$data = "$name,$age,$address,$number,$email\n";

print $WRITEFILE "$data";

close $WRITEFILE;

Tkx::tk\_\_\_messageBox(-title=>"Conformation",-message => "Your Record has been Entered Successfully!!!");

}

sub clear {

$name='';

$age='';

$address='';

$number='';

$email='';

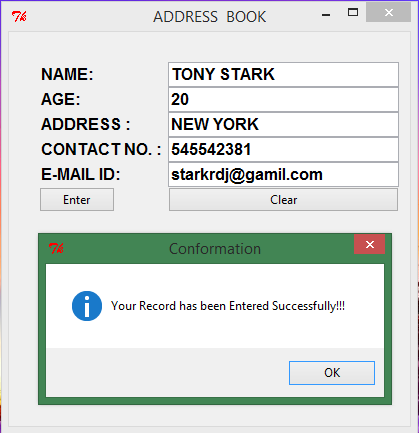
Tkx::tk\_\_\_messageBox(-title=>"Information",-message => "Book Cleared Successfully!!!",);

}

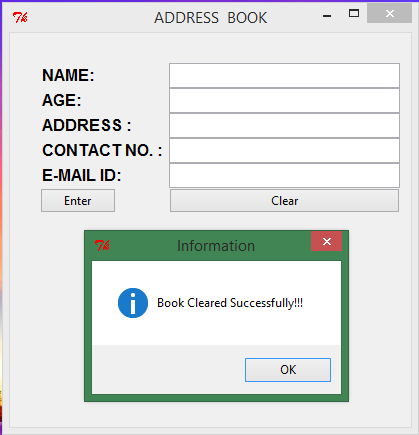
Tkx::MainLoop();

SNAPSHOTS OF OUTPUT:

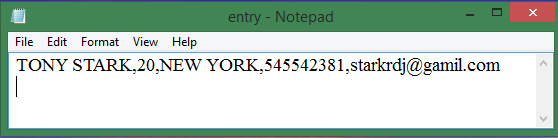
**AFTER CLICKING “ENTER” BUTTON:**



**AFTER CLICKING “CLEAR” BUTTON:**



**“ENTRY” FILE CREATED:**



**CONCLUSION:**

Thus, we have studied and implemented **GUI** in Perl. Also, we have studied the **Tkx** module with its **widgets** in Perl .