Lecture 10

GUI Programming

Prof. Steven Ludtke
N410, sludtke@bcm.edu

GUI Programming

- Tkinter, PyQt, PyGTK, wxPython, FXPy

- widget - A graphical object, like a button or a slider
- callback - a function which is called when the user interacts with a widget
- geometry or layout manager - controls where widgets are displayed

Tkinter

- ‘standard’ Python GUI toolkit
- Python interface elegant, but built on top of Tcl/Tk
- A bit clunky and slow, but has been used to build some very large applications (eg - Chimera)
- If you have a choice, use PyQt4 (personal suggestion)
- http://www.pythonware.com/library/tkinter/introduction/index.htm
- Extended by PMW and Tix

tkFileDialog

- import tkFileDialog
  - askdirectory(**options)
  - askopenfile(mode='r', **options)
  - askopenfilename(**options)
  - askopenfilenames(**options)
  - askopenfiles(mode='r', **options)
  - asksaveasfile(mode='w', **options)
  - asksaveasfilename(**options)

tkMessageBox

- import tkMessageBox
  - askokcancel(title=None, message=None, **options)
  - askquestion(title=None, message=None, **options)
  - askretrycancel(title=None, message=None, **options)
  - askyesno(title=None, message=None, **options)
  - showerror(title=None, message=None, **options)
  - showinfo(title=None, message=None, **options)
  - showwarning(title=None, message=None, **options)

tkColorChooser

- import tkColorChooser
  - askcolor(color=None, **options)

Tkinter

- Event driven programming
  - Set up all of your widgets
  - Call the event loop
  - Cleanup

```
root = Tk()                    # Initializes Tkinter
setup widgets
root.mainloop()                # Runs the GUI until the user triggers an exit
root.destroy()                 #  Cleanup
```

Simple Tkinter

```
from Tkinter import *

root = Tk()

w = Label(root, text="Hello, world!")
w.pack()

root.mainloop()
```

Tkinter Widgets

- BitmapImage
- Button
- Canvas
  - Arc, Bitmap, Image, Line, Oval, Polygon, Rectangle, Text
- Checkbutton
- Entry
- Font
- Frame (window)
- Label
- Listbox
- Menu/Menubutton
- Message
- PhotoImage
- Radiobutton
- Scale
- Scrollbar
- Text
- Toplevel Widget

Tkinter Misc

- DoubleVar
- IntVar
- StringVar

- Grid Geometry Manager
- Pack Geometry Manager

- Place Geometry Manager
- MessageBox
- SimpleDialog

- tkFileDialog
- tkColorChooser
- tkMessageBox

Full Tkinter Example

```
from Tkinter import *

class Demo(Frame):
    def say_hi(self):
        print "hi there, everyone!"

    def createWidgets(self):
        self.QUIT = Button(self)
        self.QUIT["text"] = "QUIT"
        self.QUIT["fg"]   = "red"
        self.QUIT["command"] =  self.quit

        self.QUIT.pack({"side": "left"})

        self.hi_there = Button(self)
        self.hi_there["text"] = "Hello",
        self.hi_there["command"] = self.say_hi

        self.hi_there.pack({"side": "left"})

    def __init__(self, master=None):
        Frame.__init__(self, master)
        self.createWidgets()
        self.pack()

root = Tk()
app = Demo(master=root)
app.mainloop()
root.destroy()
```

tkinter References

- http://www.pythonware.com/library/tkinter/introduction/index.htm
- http://infohost.nmt.edu/tcc/help/pubs/tkinter.pdf
- http://www.amazon.com/Python-Tkinter-Programming-Grayson-Ph-D/dp/1884777813

Web Scripting

Scripting, Server vs. Client

- Serverside scripting depends on the webserver you use
    - Many choices
    - May put load on server
- Clientside
    - Java - often available, but many issues
    - Flash - Almost ubiquitous, but somewhat proprietary
    - Javascript built in to mostbrowsers
        - AJAX - Asynchronous Javascript And XML

Javascript - Button

```
<HTML><HEAD><TITLE>Hi there</TITLE></HEAD>
<BODY>
<h3>Here is a title</h3>
And some text
<p>
<input type="button" value="Push Me" onclick="alert('You pushed me too far')">
</p>
</body>
```

Javascript - mouseover

```
<HTML><HEAD><TITLE>Hi there</TITLE></HEAD>
<BODY>
<h3>Here is a title</h3>
And some text
<p>
<a href="index3.html" onmouseover="window.document.bgColor='red'">Red</a>
<a href="index3.html" onmouseover="window.document.bgColor='green'">Green</a>
<a href="index3.html" onmouseover="window.document.bgColor='blue'">Blue</a>
<a href="index3.html" onmouseover="window.document.bgColor='white'">White</a>
</p>
</body>
```

Javascript Calculator

```
<HTML><HEAD><TITLE>Hi there</TITLE></HEAD>
<BODY>
<h3>Calculator</h3>
<form name=calc onsubmit=compute()>
<input type=text name=data></input>
</form>
<script>
document.calc.data.value=window.location.search.split("=")[1]
function compute() {
document.calc.data.value=eval(document.calc.data.value);
}
</script>
</body>
```

Javascript - Calculator #2

```
<HTML><HEAD><TITLE>Hi there</TITLE></HEAD>
<BODY>
<h3>Calculator</h3>
<form name=calc onsubmit=compute()>
<input type=text name=data value="0"></input>
<table><tr>
<td><input type="button" value="7" onclick="num('7')"></td>
<td><input type="button" value="8" onclick="num('8')"></td>
<td><input type="button" value="9" onclick="num('9')"></td>
<td><input type="button" value="X" onclick="fn('*')"></td></tr><tr>
<td><input type="button" value="4" onclick="num('4')"></td>
<td><input type="button" value="5" onclick="num('5')"></td>
<td><input type="button" value="6" onclick="num('6')"></td>
<td><input type="button" value="-" onclick="fn('-')"></td></tr><tr>
<td><input type="button" value="1" onclick="num('1')"></td>
<td><input type="button" value="2" onclick="num('2')"></td>
<td><input type="button" value="3" onclick="num('3')"></td>
<td><input type="button" value="+" onclick="fn('+')"></td></tr><tr>
<td colspan=3><input type="button" value="0" onclick="num('0')"></td>
<td><input type="button" value="=" onclick="eql()"></td>
</tr> </table> </form>
```

Javascript - Calculator #2

```
<script>
xpr=""
rst=1
function num(val) {
    xpr+=val
    if (rst) {
        rst=0
        document.calc.data.value=""
    }
    document.calc.data.value+=val
}

function fn(val) {
    xpr+=val
    rst=1
}

function eql() {
    document.calc.data.value=eval(xpr)
    xpr=""
    rst=1
}
</script>
</body>
```

Javascript - Statements

- var name[=value],name[=value]
- function f(x,y) statement
- if (expression) statement; else statement;
- do statement while (expression)
- while (expression) statement
- for (var in array) statement
- for (init; update; test) statement
- switch (expr) {

  case const:

  statements

  break

  default:

  statements }

Javascript - Events

- onclick
- onfocus, onblur
- onmousedown, up, move, over,out
- onkeydown, up, press
- onreset
- onsubmit
- onload, unload

References

- http://www.w3.org/TR/html4/
- http://www.w3.org/TR/html4/index/elements.html
- http://htmlhelp.com/reference/html40/olist.html

- http://www.javascriptkit.com/jsref
- http://www.w3schools.com/jsref/default.asp

Homework 10

- Take the program from homework 7 and add a basic GUI to it. When you run the program it should open a window with at least 3 buttons. One should open a file dialog and allow the user to select a file to be read in. The other should display a histogram of the y values using matplotlib. The third should exit the program. Feel free to add additional buttons if you like but 3 is all that is required.
- This homework is not due until next THURSDAY.
- DON'T FORGET TO WORK ON YOUR CLASS PROJECTS