İSTANBUL
ŞEHİR
UNIVERSITY

ENGR 212:
Programming Practice

Week 4

Graphical User Interface (GUI) Programming

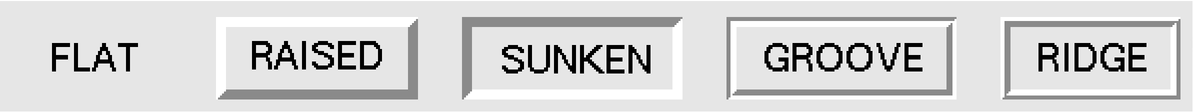
Events

Grid Geometry Manager

```python
from Tkinter import *

root = Tk()

colours = ['red','green','orange','white','yellow','blue']

r = 0
for c in colours:
    Label(root, text=c).grid(row=r,column=0, sticky=E)
    Entry(root, bg=c).grid(row=r,column=1, sticky=EW)
    r = r + 1

root.mainloop()
```

Frame Widget

```
# frame 1
frame1 = Frame(self, borderwidth=2, relief=GROOVE)
frame1.pack(side=LEFT, padx=30, pady=30)

# frame 2
frame2 = Frame(self, borderwidth=2, relief=GROOVE)
frame2.pack(side=LEFT, padx=10, pady=10)

# frame 3
frame3 = Frame(self, borderwidth=2, relief=GROOVE)
frame3.pack(side=RIGHT, padx=5, pady=5)

# Adjust the labels
Label(frame1, text="Frame 1").pack(padx=10, pady=10)
Label(frame2, text="Frame 2").pack(padx=10, pady=10)
Label(frame3, text="Frame 3").pack(padx=10, pady=10)
```



Relief styles

Events

- An ***event*** is some occurrence that your application needs to know about.
- An ***event handler*** is a function in your application that gets called when an event occurs.
- We call it ***binding*** when your application sets up an event handler that gets called when an event happens to a widget.

Events

- Event Sources:
  - Mouse Operations by user
  - Key Presses by user
  - Redraw Events by window manager
- Capturing and handling events

```
widget.bind(event, handler)
```

- If an event matching the *event* description occurs on the widget, *handler* is called with an event object

Events – Capturing Clicks

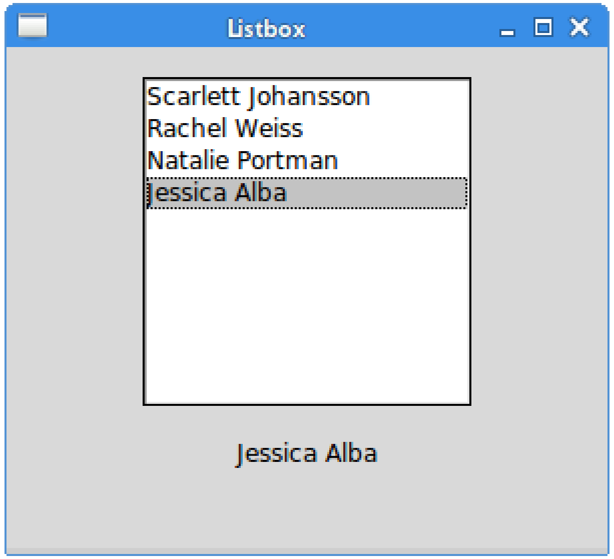
```
from Tkinter import *

root = Tk()

def callback(event):
    print "clicked at", event.x, event.y

frame = Frame(root, width=100, height=100)
frame.bind("<Button-1>", callback)
frame.pack()

root.mainloop()
```

Events – Capturing Clicks

```
from Tkinter import *

root = Tk()

colors = ['red', 'yellow', 'green']
color_index = 0

def onClick(event):
    global color_index
    widget = event.widget
    color_index = (color_index+1) % len(colors)
    widget.config(bg = colors[color_index])
    print "clicked at", event.x, event.y

frame = Frame(root, width=100, height=100)
label = Label(frame, text = 'Click me to change my color!')
label.bind("<Button-3>", onClick)
label.pack()
frame.pack()

root.mainloop()
```

Events – Capturing ListBox Selections

```
class Example(Frame):

    def __init__(self, parent):
        Frame.__init__(self, parent)
        self.parent = parent
        self.initUI()

    def initUI(self):
        self.pack(fill=BOTH, expand=1)
        acts = ['Scarlett Johansson', 'Rachel Weiss',
            'Natalie Portman', 'Jessica Alba']
        lb = Listbox(self)
        for i in acts:
            lb.insert(END, i)

        lb.bind("<<ListboxSelect>>", self.onSelect)
        lb.pack(pady=15)
        self.var = StringVar()
        self.label = Label(self, text=0, textvariable=self.var)
        self.label.pack()

    def onSelect(self, val):
        sender = val.widget
        idx = sender.curselection()
        value = sender.get(idx)
        self.var.set(value)

def main():
    root = Tk()
    ex = Example(root)
    root.geometry("300x250+300+300")
    root.mainloop()

main()
```



Scarlett Johansson
Rachel Weiss
Natalie Portman
Jessica Alba

Jessica Alba

List of Events

- <Button-1> <Button-2> <Button-3> <Button-4> <Button-5>
- <Motion>
- <ButtonRelease-1> <ButtonRelease-2> <ButtonRelease-3>
- <Double-Button-1> <Double-Button-2> <Double-Button-3>
- <Enter>
- <Leave>
- <FocusIn>
- <FocusOut>
- <Return>
- <Key>
- a
- <Shift-Up>
- <Configure>

Tkinter References

- http://www.tutorialspoint.com/python/python_gui_programming.htm
- http://zetcode.com/gui/tkinter/
- http://effbot.org/tkinterbook/

Collective Intelligence

Introduction

- Netflix is an online DVD rental company.
- Netflix lets people choose movies to be sent to their homes, and makes recommendations based on the movies that customers have previously rented.
- By using data about which movies each customer enjoyed, Netflix is able to recommend movies to other customers that they may never have even heard of and keep them coming back for more.
- Any way to improve its recommendation system is worth a lot of money to Netflix.

Introduction

- 2006: announcement of a prize of $1 million to the first person to improve the accuracy of its recommendation system by 10 percent.
- Thousands of teams from all over the world.
- The leading team improved 7 percent within 1st year.

Introduction

- Google started in 1998, when there were Netscape, AOL, Yahoo, MSN.
- Google took a completely new approach to ranking search results by using the links on millions of web sites to decide which pages were most relevant.
- Google’s search results were so much better than those of the other players that by 2004 it handled 85 percent of searches on the Web.

What do these companies have in common?

- Please elaborate...

What do these companies have in common?

- The ability to collect information
  - Sophisticated algorithms: ***combine data collected from many different sources***.
- The computational power to interpret it
  - enabling great collaboration opportunities and a better understanding of users/customers (**Data Mining**).
  - Everyone wants to understand their customers better in order to create more <u>targeted advertising</u>.

Combine data

- Machine learning
  - a subfield of artificial intelligence (AI)
  - focuses on algorithms that allow computers to “learn”.
  - in most cases:
    - an algorithm is given a set of data and infers **information** about the properties of the data.

Combine data

- What do we do with that information?
  - make **predictions** about other data that it might see in the *future*.
- How?: Patterns!
  - Almost all nonrandom data contains *patterns*
  - Patterns allow the machine to generalize.
- How does it generalize?
  - builds and trains a **model**
  - determines the **important aspects of the data**.

Real-Life Examples

- **Google** not only uses web links to rank pages
  - it constantly gathers information on when advertisements are clicked by different users
    - target the advertising more effectively.
- Sites like **Amazon** and **Netflix** use information about the things people buy or rent to
  - determine similar people or items
  - make recommendations based on purchase history.

Real-Life Examples

- **Pandora** and **Last.fm**
  - use your ratings of different bands and songs
  - create custom radio stations with music they think you will enjoy.
- Some dating sites, such as **eHarmony**
  - use information collected from participants
  - determine who would be a good match.

Time to think..!

- Spend several minutes now to find out what great information **service** you can build.
- Do a web search or day-dream, and transcend into the knowledge world.
- Go!

Part 2: Making Recommendations

Amazon.com

- Amazon tracks the purchasing habits of all its shoppers
  - when you log onto the site, it uses this information to suggest products you might like.
- Amazon can even suggest *movies* you might like, even if you've only bought **books** from it before.

Collaborative Filtering

- A collaborative filtering algorithm usually works by
  - searching a large group of people, and
  - finding a smaller set with tastes **similar** to yours.
  - It looks at other things they like and combines them to create a ranked list of suggestions.

Preferences

- # Dictionary of movie critics and their ratings of movies

```
critics={
    'Lisa Rose': { 'Lady in the Water': 2.5,
                   'Snakes on a Plane': 3.5,
                   'Just My Luck': 3.0,
                   'Superman Returns': 3.5,
                   'You, Me and Dupree': 2.5,
                   'The Night Listener': 3.0},
    'Gene Seymour': { 'Lady in the Water': 3.0,
                               ...},
    ...
```

Preferences

- # Dictionary of movie critics and their ratings of movies

```
critics={
    'Lisa Rose': { 'Lady in the Water': 2.5,
                   'Snakes on a Plane': 3.5,
                   'Just My Luck': 3.0,
                   'Superman Returns': 3.5,
                   'You, Me and Dupree': 2.5,
                   'The Night Listener': 3.0},
    'Gene Seymour': { 'Lady in the Water': xx,
                               ...},
    ...
```

Critic

Preferences

- # Dictionary of movie critics and their ratings of movies

```
critics={
    'Lisa Rose': { 'Lady in the Water': 2.5,
                   'Snakes on a Plane': 3.5,
                   'Just My Luck': 3.0,
                   'Superman Returns': 3.5,
                   'You, Me and Dupree': 2.5,
                   'The Night Listener': 3.0},
    'Gene Seymour': { 'Lady in the Water': xx,
                                ...},
    ...
```

Movie

Preferences

- # Dictionary of movie critics and their ratings of movies

```
critics={
    'Lisa Rose': { 'Lady in the Water': 2.5,
                   'Snakes on a Plane': 3.5,
                   'Just My Luck': 3.0,
                   'Superman Returns': 3.5,
                   'You, Me and Dupree': 2.5,
                   'The Night Listener': 3.0},
    'Gene Seymour': { 'Lady in the Water': xx,
                               ...},
    ...
```

Rating: 1-5

Week 4/code.zip

- Go to LMS and download code.zip (Lectures/Week4/Code.zip)
- Unzip recommendations.py
- Load the Python module recommendations.py

```
 from recommendations import *
```

- and play around with `critics` dictionary (i.e., dataset).