CS 580 Client-Server Programming
Spring Semester, 2006
Doc 23 Some Ruby GUI
Apr 25, 2006

References

Progamming Ruby 2'nd Ed, Thomas, Pragmatic Bookshelf, 2005

Mastering Perl/Tk, Lidie & Walsh, O'Reilly, 2002

Ruby/TK Tutorial
http://members.chello.nl/~k.vangelder/ruby/learntk/

Documentation?

Books on Perl Tk

Ruby/TK Tutorial

http://members.chello.nl/~k.vangelder/ruby/learntk/

Google Ruby Tk

Hello World

```
require 'tk'
root = TkRoot. new { title "HW Example"}
TkLabel.new(root) { text 'Hello'; pack}
Tk.mainloop
```

Variations

```
require 'tk'
root = TkRoot. new { title "HW Example"}
TkLabel.new(root) { text('Hello, World'); pack('side'=>'left', 'padx'=>10, 'pady'=>10)}
Tk.mainloop
```

```
require 'tk'
root = TkRoot. new { title "HW Example"}
TkLabel.new(root) do
    text('Hello, World')
    pack('side'=>'left', 'padx'=>10, 'pady'=>10)
end
Tk.mainloop
```

```
require 'tk'
TkLabel.new do
    text('Hello, World')
    pack('side'=>'left', 'padx'=>10, 'pady'=>10)
end
Tk.mainloop
```

Widgets

Button
Checkbutton
Radiobutton
Label
Entry
Scrollbar
Lstbox
Text
TextUndo
ROText
Canvas
Scale
Frame
MainWindow
TopLevel
Menu

Exit and Destroy


```
require 'tk'
TkButton.new do
    text('Exit')
    pack('side'=>'left', 'padx'=>10, 'pady'=>10)
    command {exit}
end
Tk.mainloop
```

```
require 'tk'
TkButton.new do
    text('Exit')
    pack('side'=>'left', 'padx'=>10, 'pady'=>10)
    command {root.destroy}
end
Tk.mainloop
```

exit

Only use in file containing require 'tk'

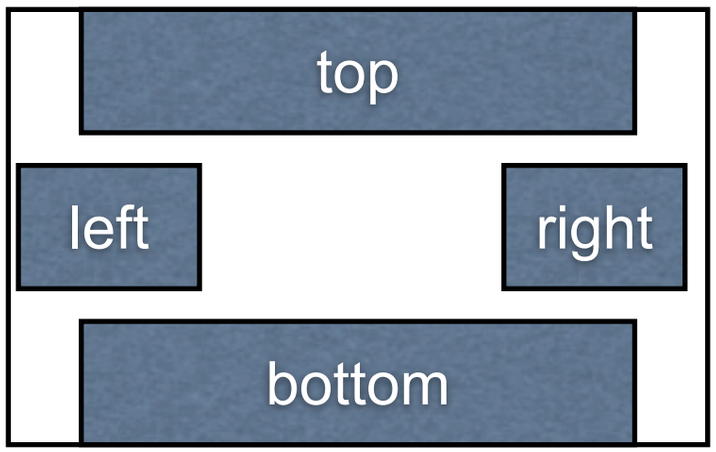
No clean up possible

Destroy Example

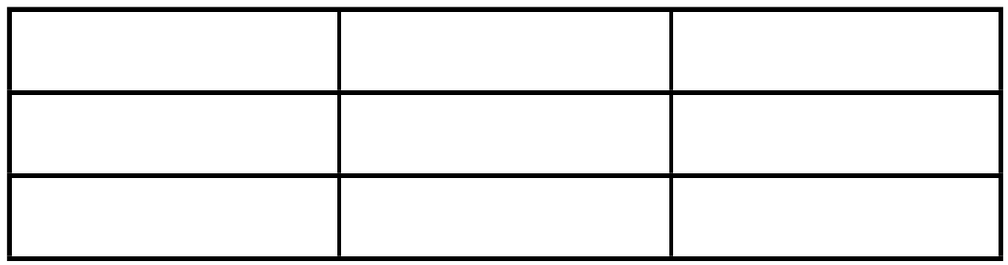
```
require 'tk'
class SimpleApplication

  def foo
    @text.value = @text.value + 'foo'
  end

  def bar
    @text.value = @text.value + 'bar'
  end

  def initialize
    pading = { 'padx' => 10, 'pady' => 10 }
    doFoo = proc {foo}
    root = TkRoot.new { title "Simple" }
    @text = TkVariable.new
    @entry = TkEntry.new(root, 'textvariable' => @text)
    @entry.pack(pading)
    TkButton.new(root) {text 'Do It'; command doFoo; pack pading}
    doIt2 = TkButton.new(root)
    doIt2.text('Do it 2')
    doIt2.command(bar)
    doIt2.pack(pading)
    TkButton.new(root) {text 'Exit'; command {root.destroy}; pack pading}
  end

  def show
      Tk.mainloop
      puts 'Clean up'
  end
end

SimpleApplication.new.show
puts 'the end'
```

Geometry Managers

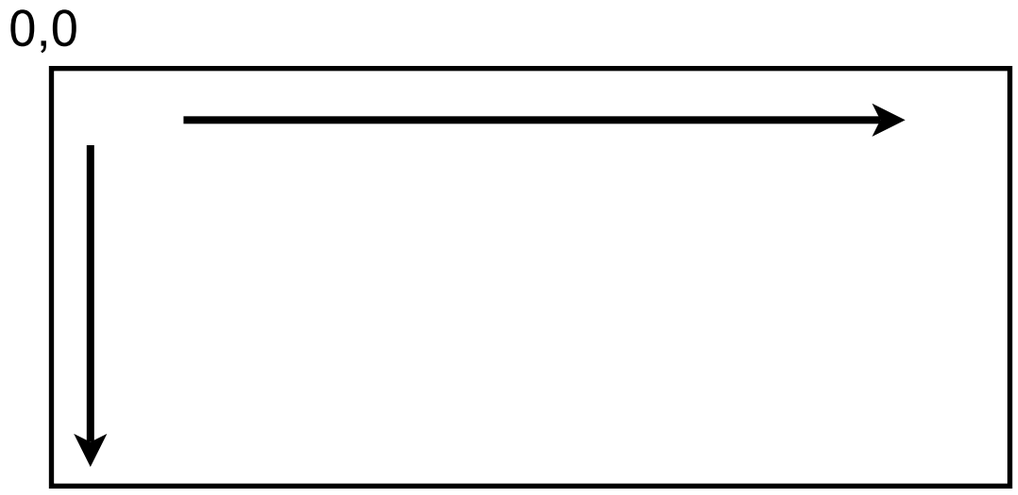
pack



grid



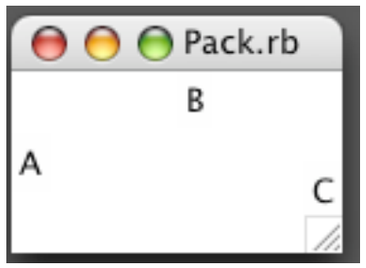
place



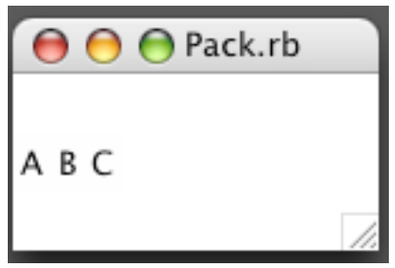
Pack Options

side => 'left' | 'right' | 'top' 'bottom'	location of widget
fill => 'none' | 'x' | 'y' | 'both'	widget fills allocation rect
expand => 1 | 0	allocation rect fill remaining space
anchor => 'n' | 'ne' | 'e' | 'se' | 's' | 'sw' | 'w' | 'nw' | 'center'	anchors widget in allocation rect
after => otherWidget	order of widget
before => otherWidget	
ipadx => amount	increase widget size by amount
ipady => amount	
padx => amount	Add padding to widget
pady => amount	

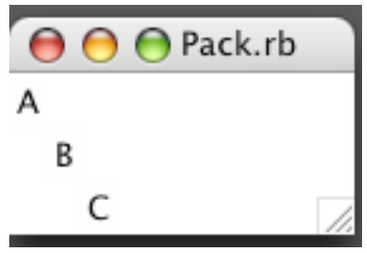
Pack Examples




```
require 'tk'
TkLabel.new do
    text('A')
    pack('side'=>'left')
end
TkLabel.new do
    text('B')
    pack('side'=>'top')
end
TkLabel.new do
    text('C')
    pack('side'=>'right')
end
Tk.mainloop
```

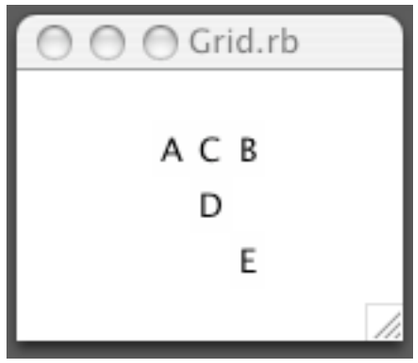



```
require 'tk'
TkLabel.new do
    text('A')
    pack('side'=>'left')
end
TkLabel.new do
    text('B')
    pack('side'=>'left')
end
TkLabel.new do
    text('C')
    pack('side'=>'left')
end
Tk.mainloop
```




```
require 'tk'
TkLabel.new do
    text('A')
    pack('side'=>'left','anchor'=>'n')
end
TkLabel.new do
    text('B')
    pack('side'=>'left' , 'anchor'=>'center')
end
TkLabel.new do
    text('C')
    pack('side'=>'left', 'anchor'=>'sw')
end
Tk.mainloop
```

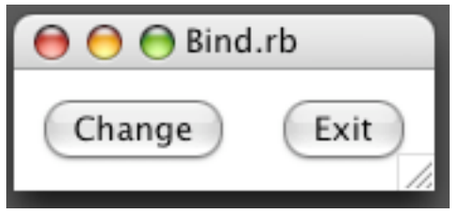
Grid




```
require 'tk'

TkLabel.new do
    text('A')
    grid('row'=>0, 'column'=> 0)
end
TkLabel.new do
    text('B')
    grid('row'=>0, 'column'=> 3)
end
TkLabel.new do
    text('C')
    grid('row'=>0, 'column'=> 2)
end
TkLabel.new do
    text('D')
    grid('row'=>2, 'column'=> 2)
end
TkLabel.new do
    text('E')
    grid('row'=>3, 'column'=> 3)
end
Tk.mainloop
```

Events

```
require 'tk'
change = TkButton.new do
    text('Change')
    pack('side'=>'left', 'padx'=>10, 'pady'=>10)
end

change.bind("Enter") {change.configure('text'=>'Enter')}
change.bind("Leave") {change.configure('text'=>'Leave')}
change.bind("Motion") {change.configure('text'=>'Going')}

TkButton.new do
    text('Exit')
    pack('side'=>'left', 'padx'=>10, 'pady'=>10)
    command {exit}
end
Tk.mainloop
```

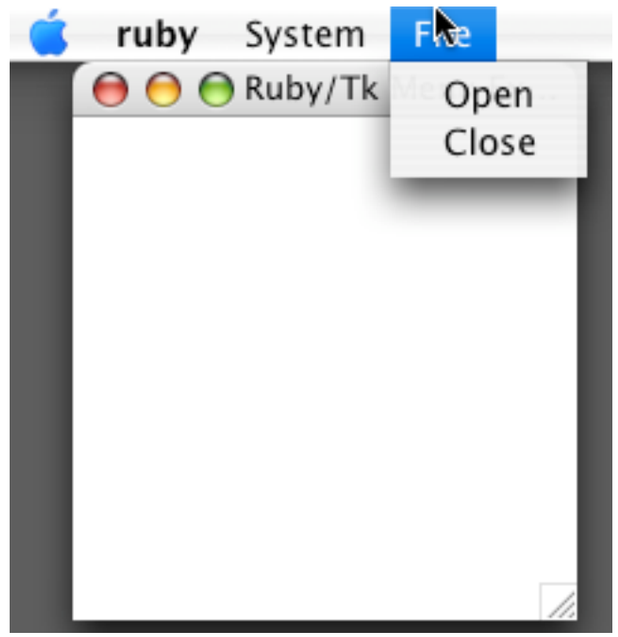


Passwords

```
require 'tk'
class PasswordExample

  def passwordStartsWithA?
    @password.value[0,1] == 'a'
  end

  def initialize
    pading = { 'padx' => 10, 'pady' => 10 }
    checkPassword = proc {passwordStartsWithA?}
    root = TkRoot.new { title "Login" }
    @password = TkVariable.new
    @text = TkVariable.new
    passwordFrame = TkFrame.new(root) {pack()}
    TkLabel.new(passwordFrame) do
      text('Password')
      grid('column'=>0, 'row'=>0)
    end
    TkEntry.new(passwordFrame,
                  'textvariable' => @password) do
      validate('focusout')
      validatecommand(checkPassword)
      invalidcommand(lambda {focus; bell})
      grid('column'=>1, 'row'=>0)
      show('*')
    end
    TkLabel.new(passwordFrame) do
        text('Username')
        grid('column'=>0, 'row'=>1)
      end
      TkEntry.new(passwordFrame, 'textvariable' => @text) do
          textvariable( @text)
          grid('column'=>1, 'row'=>1)
          show('*')
      end

      TkButton.new(root) {text 'Exit';
                  command {root.destroy};
                  pack pading}
    end

end

PasswordExample.new
Tk.mainloop
```

Menu Style 1

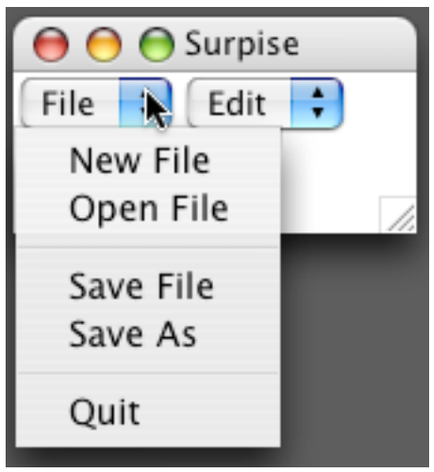
```
require 'tk'
root = TkRoot.new() { title "Ruby/Tk Menu Example" }

bar = TkMenu.new()

sys = TkMenu.new(bar)
sys.add('command', 'label'=>"Quit", 'command'=>proc { root.destroy })
bar.add('cascade', 'menu'=>sys, 'label'=>"System")

file = TkMenu.new(bar)
file.add('command', 'label'=>"Open", 'command'=>proc { puts "Open..." })
file.add('command', 'label'=>"Close", 'command'=>proc { puts "Close..." })
bar.add('cascade', 'menu'=>file, 'label'=>"File")

root.menu(bar)
Tk.mainloop
```



http://members.chello.nl/~k.vangelder/ruby/learntk/

Menu Style 2

```
require 'tk'
menu_spec = [
  [ ['File', 0],
    ['New File',  proc{puts 'New File'}],
    ['Open File', proc{puts 'Open File'}],
    '---',
    ['Save File', proc{puts 'Save File'}],
    ['Save As',   proc{puts 'Save As'}],
    '---',
    ['Quit',      proc{exit}]
  ],
  [ ['Edit', 0],
    ['Cut',       proc{puts 'Cut'}],
    ['Copy',      proc{puts 'Copy'}],
    ['Paste',     proc{puts 'Paste'}]
  ]
]
root = TkRoot.new {title "Surpise"}
TkMenubar.new(root, menu_spec, 'tearoff'=>false).pack('fill'=>'x', 'side'=>'top')
Tk.mainloop
```



http://members.chello.nl/~k.vangelder/ruby/learntk/