Lecture 18

Chapter 15 Event-driven Programming

Textbox

15.4.1 Adding and Moving Shapes on a Canvas

Fibonacci Numbers (HW 10)

Example 1: using TextBox

Let's write a program that allows us type in something in a text box and then retrieve it.

See textBoxExample1.py and textBoxExample2.py

Example 2: Adding and Moving Shapes on a Canvas

Let's write a program that allows us to create and modify shapes.

- Each time the mouse is clicked on the background – a new shape is created.

- Dragging a mouse on an existing shape is used to move it.

- Pressing a key on the keyboard changes color of the shape our mouse is over (*r* will be for 'red', *b* will be for 'blue', ...)

Example 2: Adding and Moving Shapes on a Canvas

Let's think:

1) What figures we'd like to play?
   circle, rectangle, triangle, star, hexagon

2) How many handlers do we need and what will they be responsible for?

Example 2: Adding and Moving Shapes on a Canvas

Let's think:

1) What figures we'd like to play?
circle, rectangle, triangle, star, hexagon

2) How many handlers do we need and what will they be responsible for?

- Exit Button Handler
- Mouse Event or Shape Handler – let's split it to two
  - ShapeHandler
  - NewShapeHandler

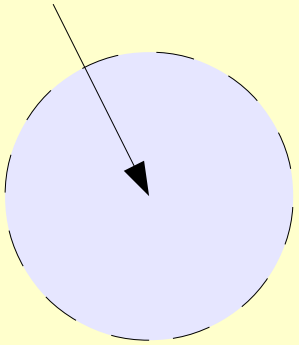
Example 2: Adding and Moving Shapes on a Canvas

★ NewShapeHandler (newShape)

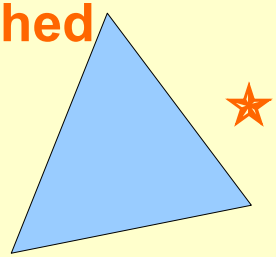
HELP

Click on the paper:
new shape is created

★ MenuButtonHandler
(exitHandler)



★ ShapeHandler
is attached



★ ShapeHandler
is working

Click on the existing
shape: can drag it

EXIT

★ ExitButtonHandler
(exitHandler)

See ShapesDragging.py and ShapesWork,py

Fibonacci Numbers

From HW10:

To calculate Fibonacci(6) is very wasteful:

Fibonacci(4) is calculated 2 times

Fibonacci(3) is calculated 3 times

Fibonacci(2) is calculated 5 times

Fibonacci(1) is calculated 3 times

Fibonacci Numbers

A better recursive version for calculating Fibonacci numbers:

```
def newFib(n):
    return newFib2(1, 1, 0, n)

def newFib2(curr, prev, i, n):
    if i == n - 2:
        return curr
    else:
        return newFib2(curr + prev,curr, i + 1, n)
```

Note: definition won't work if a user wants to get the first Fibonacci number, i.e. 1, but it works perfectly well for all other cases.