Lecture 9

Topics: *Chapter 4. Objects and Graphics*

4.4 Using Graphical Objects (continues)
4.5 Graphing Future Value
4.7.1 Graphing Mouse Clicks

4.4 Using Graphical Objects

Caution: It is possible for two different variables to refer to exactly the same object. Changes made to the object through one variable are visible to the other one.

4.4 Using Graphical Objects

Caution: It is possible for two different variables to refer to exactly the same object. Changes made to the object through one variable are visible to the other one.

Example: Let’s draw two ovals of the same size at two different places:

4.4 Using Graphical Objects

Caution: It is possible for two different variables to refer to exactly the same object. Changes made to the object through one variable are visible to the other one.

Example: Let’s draw two ovals of the same size at two different places:

```
window = GraphWin('Drawing two ovals',640,480)

left_o = Oval(Point(50,50),Point(90,70))
left_o.setOutline('orange')
left_o.setFill('yellow')

right_o = left_o
right_o.move(160,0)

left_o.draw(window)
right_o.draw(window)
```

see ovals.py

4.4 Using Graphical Objects

```
Traceback (most recent call last):
  File
"/Users/luis/teaching/classes/22-1/csi31/myslides/Lecture09/ovals.py
", line 25, in <module>
    main()
  File
"/Users/luis/teaching/classes/22-1/csi31/myslides/Lecture09/ovals.py
", line 19, in main
    right_o.draw(window) # drawing right oval
  File
"/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.1
0/site-packages/graphics.py", line 481, in draw
    if self.canvas and not self.canvas.isClosed(): raise
GraphicsError(OBJ_ALREADY_DRAWN)
graphics.GraphicsError: Object currently drawn
```

4.4 Using Graphical Objects

Caution: It is possible for two different variables to refer to exactly the same object. Changes made to the object through one variable are visible to the other one.

Example: Let’s draw two ovals of the same size at two different places:

```
window = GraphWin('Drawing two ovals',640,480)

left_o = Oval(Point(50,50),Point(90,70))
left_o.setOutline('orange')
left_o.setFill('yellow')

right_o = left_o
right_o.move(160,0)

left_o.draw(window)
right_o.draw(window)
```

crashes here

see ovals.py

4.4 Using Graphical Objects

Caution: It is possible for two different variables to refer to exactly the same object. Changes made to the object through one variable are visible to the other one.

Example: Let's draw two ovals of the same size at two different places:

```
window = GraphWin('Drawing two ovals',640,480)

left_o = Oval(Point(50,50),Point(90,70))
left_o.setOutline('orange')
left_o.setFill('yellow')

right_o = left_o
right_o.move(160,0)

left_o.draw(window)
right_o.draw(window)
```

problem is here

crashes here

see ovals.py

4.4 Using Graphical Objects

How to correct the problem?

4.4 Using Graphical Objects

How to correct the problem?

1. write a separate code for the right oval, or

2. clone the first one (method `clone()`) see ovals-corrected.py

4.4 Using Graphical Objects

Caution: It is possible for two different variables to refer to exactly the same object. Changes made to the object through one variable are visible to the other one.

Example: Let's draw two ovals of the same size at two different places:

```
window = GraphWin('Drawing two ovals',640,480)

left_o = Oval(Point(50,50),Point(90,70))
left_o.setOutline('orange')
left_o.setFill('yellow')

right_o = left_o.clone()
right_o.move(160,0)

left_o.draw(window)
right_o.draw(window)
```

$\longleftarrow$ problem is fixed

see ovals-corrected.py

4.5 Graphing Future Value

It is difficult to make a budget that spans several years, because prices are not stable. If your company needs 200 pencils per year, you cannot simply use this year's price as the cost of pencils two years from now. Because of inflation the cost is likely to be higher than it is today.

Write a program to gauge the expected cost of an item in a specified number of years. The program asks for the cost of the item, the number of years from now that the item will be purchased, and the rate of inflation. The program then outputs the estimated cost of the item after the specified period.

Have the user enter the inflation rate as a percentage, like 5.6 (%). Your program should then convert the percent to a fraction, like 0.056, and should use a loop to estimate the price adjusted for inflation.

4.5 Graphing Future Value

Design / Algorithm:

```
print an introduction
get the price
get the number of years
get the inflation rate in %, convert to decimal
repeat years times:  price = price + price*inflation
output the final price
```

see future-price-text.py

4.5 Graphing Future Value

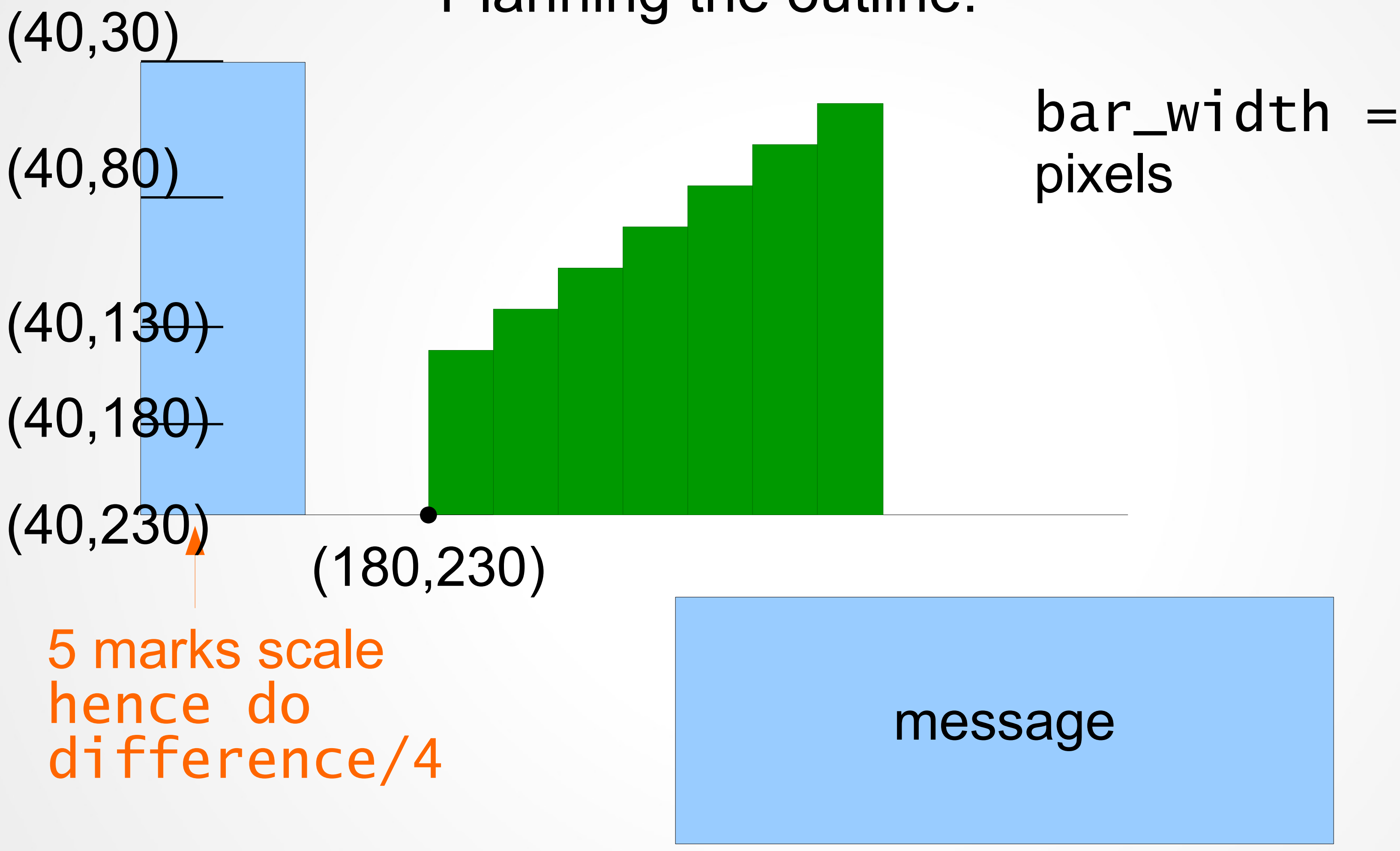
Design / Algorithm:

```
print an introduction
get the price
get the number of years
get the inflation rate in %, convert to decimal
repeat years times:  price = price + price*inflation
output the final price
```

Now let us do it in graphics!

4.5 Graphing Future Value

Planning the outline:



`bar_width = 40` pixels

see future-price_graphics.py

4.7.1 Graphing Mouse Clicks

Consider the following code:

```
win = GraphWin("Mouse Clicks",800,600)
x = win.getMouse()
x_coord = x.getX()
y_coord = x.getY()
```

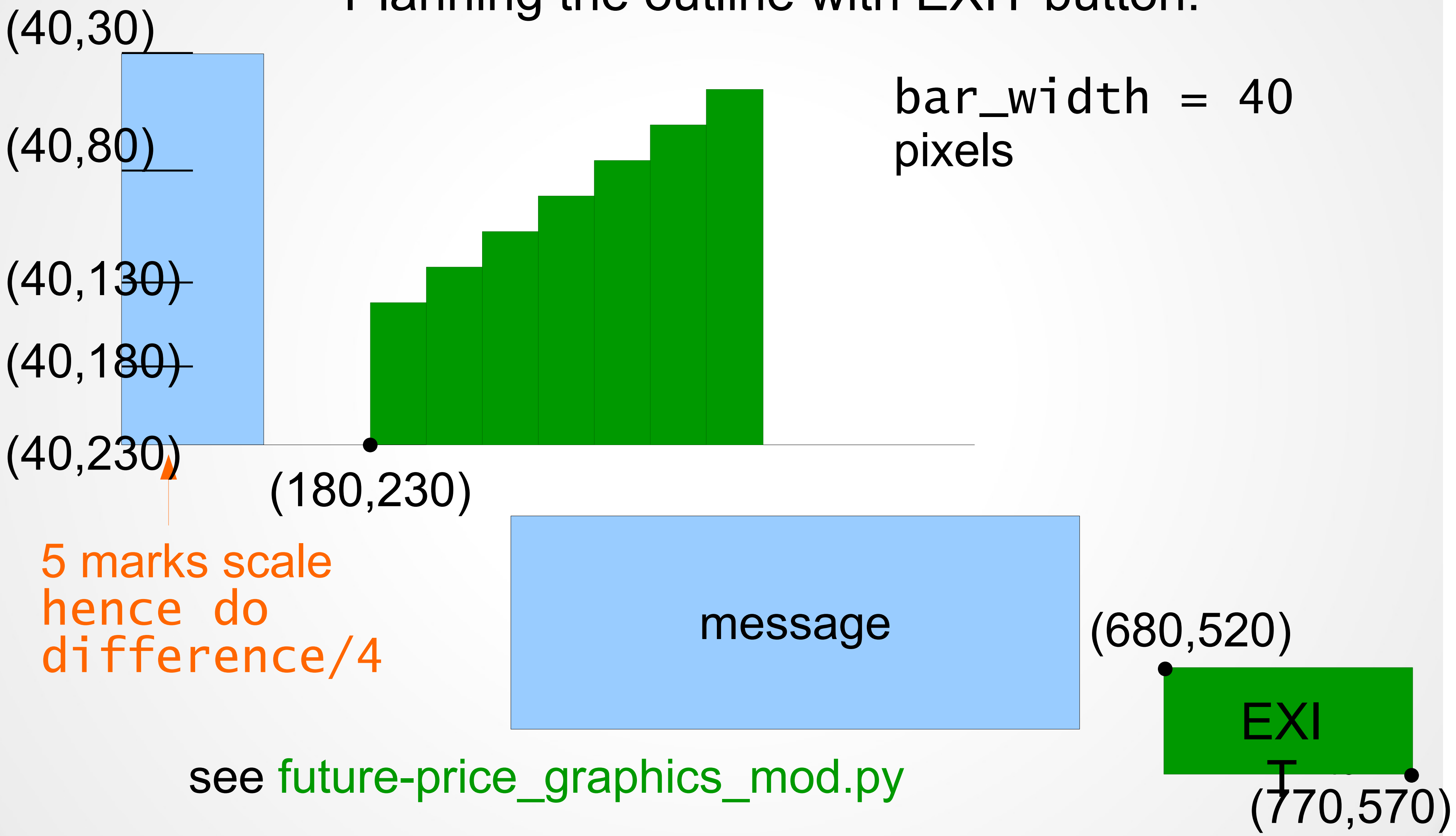
the point where the mouse click occured

the x- and y-coordinates of the point x

see mouseClicks.py

4.7.1 Graphing Mouse Clicks

Planning the outline with EXIT button:



see future-price_graphics_mod.py