CSI31 Lecture 15

Topics: *Chapter 5. Objects and Graphics*

5.6 Choosing coordinates

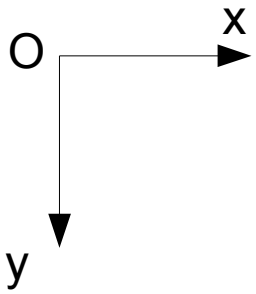
5.7 Interactive Graphics

HW №12 (due date is Monday, November 10th):

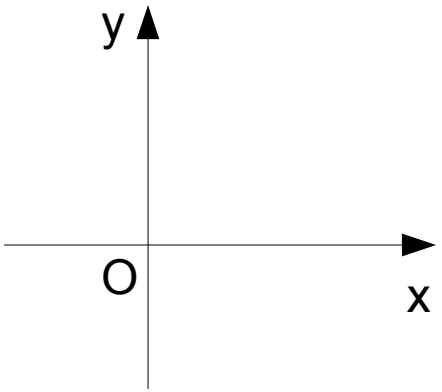
programming exersise 8, on page 161

5.6 Choosing coordinates

You probably noticed, that when we draw anything using graphics library, we have to keep in mind, that the origin is at the top left corner of the window:



i.e. it is not the usual rectangular coordinate system



So if we happen to have x- and y-values coming from rectangular coordinate system we'll have to do a conversion of coordinates.

Another issue: Imagine, that we decided to write a program of tic-tac-toe.
We'll start with drawing of 3 rows and 3 columns (i.e. two horizontal lines and two vertical lines). At this point we will have to decide what are the coordinates of the lines, depending on what are the dimensions of the window (the size).

To ease our life class `GraphWin` in the graphics library we use allows us to specify a coordinate system for the window using the `setCoords` method. The method has four parameters: the first two – the lower left corner, and the second two – the upper right corner. All subsequent drawings will be done with respect to the altered coordinate system (except for `plotPixel`)

see tic-tac-toe_begin.py

5.7 Interactive Graphics

Graphical interfaces can be used for input as well as output.

Usually in a GUI environment we can click on buttons, choose items from menus, type information into on-screen text boxes, and so on.

When the user moves the mouse, clicks on a button or types a key on the keyboards, this generates an event. This event object is further sent to an appropriate part of the program to be processed. And then the appropriate action will be taken.

Event-driven programming can be tricky for novice programmers, since it is hard to figure out «who's in charge» at any given moment. Our graphics module hides the underlying event-handling mechanism and provides two simple ways of getting user input in a `GraphWin`:

- mouse clicks
- textual input

Getting Mouse Clicks

`getMouse` method:
- when it is invoked the program pauses and waits the user to click the mouse somewhere in the graphics window,
  then the spot where the user clicks returned to the program as `Point`.

Let's write a program that returns the spot where the user clicked the mouse:
mouse-click.py

Now, let's write a program, where a program askes the user to click a mouse 4 times a draws a quadrilateral: quadrilateral.py

Textual Input

Graphics library has a simple `Entry` object, that
draws a box on the screen that can contain text.

`Entry(centerPoint,width)`,
where width is the number of characters of text that can be displayed.

It has `setText` and `getText` methods.
`setText(string)` – sets the text of the object to `string`
`getText()` - returns the current string

Let's write a program that converts centimiters to inches:
The conversion is as follows: 1 cm = 2.54 inches
see conversion.py