Introduction to Graphics
(using Zelle's 'graphics.py')

30 Nov 2009
CMPT140
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Bitsets

- Another way to implement sets is using a bitset: binary form of an integer represents flags:
- e.g., file permissions: a user may have permission to read, write, and/or execute
  - Let 4=read (r) , 2=write (w), 1=execute (x)
  - So the number 5 represents read+execute
- In Python, using bitwise shift operators:

```
readFlag = 1 << 2      # fancy way of saying 4
writeFlag = 1 << 1     # 2
execFlag = 1 << 0      # 1
```

Bitsets

- We can combine these flags using bitwise logical operators: or (|), and (&)

```
5 | 3 == 7          # (101) | (011) == (111)
5 & 3 == 1          # (101) & (011) == (001)
```

- Add read permission if not already there:

```
myPerms |= readFlag
```

- Check if we have write permission:

```
if myPerms & writeFlag:
```

- Need read perm on both file and directory:

```
if filePerms & dirPerms & writeFlag:
```

Graphics libraries

- The windowing system is the environment in which windows, desktop, etc. get placed
  - MS Windows, OS X, Linux X11, etc.
- Graphics libraries allow program code to draw objects within a window
  - Interact with the windowing system
  - For Python: Tk, PyQt, etc.
  - Zelle's graphics.py is a wrapper around Tk
- Widgets are graphical components
  - Buttons, dials, textboxes, canvas to draw

'graphics.py': Getting started

- A very rudimentary library, but quick+easy
- Download 'graphics.py' and put in current dir
- From your own *.py file, import graphics:
  - **from graphics import ***
- Create a new window object:
  - **win = GraphWin()**
- Create a point and draw it as a dot:
  - **pt = Point(100, 50)**
  - **pt.draw(win)**
- Window coordinates: (0,0) at top-left, units in pixels (but see .setCoords())

Drawing/moving a Circle

- Circle: centred about a Point, with a radius
  - **circ = Circle(pt, 20)**
  - **circ.draw(win)**
  - Or, all in one line (not saving the object):
    - **Circle(Point(100, 50), 40).draw(win)**
- Set fill colour:
  - **circ.setFill('red')**
  - Colour strings defined in graphics.py, or use color_rgb(r, g, b) to mix your own
- Move the existing circle down by 20px:
  - **circ.move(0, 20)**

Methods for all widgets

- All graphics objects created by 'graphics.py' understand the following methods:
- .draw(win): draw into the given window
- .undraw(): hide (without destroying object)
- .setFill(colour): change fill colour inside
- .setOutline(colour): change line colour
- .setWidth(pixels): change line width
- .move(dx, dy): undraw and redraw @new pos
- .clone(): deep copy

Other graphics objects

- Line: define start and end points
  - ln = Line(Point(20, 150), Point(180, 150))
  - Use .setOutline() and .setWidth() to change colour/width of the drawn line
  - .setArrow(): draw arrowheads
- Rectangle: define opposite corners
  - **Rect = Rectangle(pt1, pt2)**
- Oval: define bounding box (two Points)
- Polygon: list of vertices

Text input/output widgets

- Text output/display: define centre of textbox
  - **txt = Text(pt, “Hello, World!”)**
  - .setText() changes text string
  - .setFace(), .setStyle(), .setSize(): font face, italic/bold, font size in points
  - .setTextColor(): same as .setFill()
- Text input: specify centre and width in chars
  - **inp = Entry(Point(100, 180), 8)**
  - .setText() changes current text string
  - .getText() returns current text string

Methods on the window

- The GraphWin object is the window
  - Your program may open several windows
- Set the title and size on instantiation:
  - **win = GraphWin(“My Program!”, 50, 200)**
- Set background colour of the window:
  - **win.setBackground('white')**
- Pause and wait for user to click:
  - **win.getMouse()**
  - Returns Point where user clicked
- Shutdown window: win.close()

Changing coordinate systems

- By default, (0,0) is top-left; units are in pixels
  - If window resizes, objects do not rescale!
- Set the coordinate system of the window:
  - **win.setCoords(left, bot, right, top)**
  - Specify the coordinates of the edges of the window
  - e.g.: put (0,0) in lower left; (10,10) in T-R:
    - **win.setCoords(0, 0, 10, 10)**
  - Draw a circle in centre:
    - **Circle(Point(5,5), 4).draw()**