

Python – Módulo Graphics

André Prisco
Rafael Penna



Objetivo

- ✓ Conhecer um módulo simples para trabalhar com gráficos (janelas, pontos, desenhos, imagens, etc).

O Módulo

- Graphics.py é um simples pacote gráfico orientado à objeto, projetado para programadores iniciantes.
- Escrito por John Zelle para uso com o livro "Python Programming: An Introduction to Computer Science".
- Versão atual: <http://mcsp.wartburg.edu/zelle/python/>

O Módulo

- | Existem 2 tipos de objetos no pacote:
 - | **GraphWin**: implementa uma janela, onde desenhos podem ser feito.
 - | **GraphicsObjects**: implementa objetos que podem ser desenhados dentro de um GraphWin.
- | Exemplo simples: desenhar um círculo de raio 10 centralizado em uma janela 100x100.

```
1  from graphics import *
2  import math
3
4  def main():
5      win = GraphWin("My Circle", 100, 100)
6      c = Circle(Point(50,50), 10)
7      c.draw(win)
8      win.getMouse() # pause for click in window
9      win.close()
10  main()
11
```

O Módulo

- ‡ O módulo provê os seguintes objetos gráficos ([GraphicsObjects](#)):
 - ‡ Point
 - ‡ Line
 - ‡ Circle
 - ‡ Oval
 - ‡ Rectangle
 - ‡ Polygon
 - ‡ Text
 - ‡ Entry
 - ‡ Image.

O Módulo

- | Vários atributos dos objetos gráficos podem ser alterados:
 - | Cor da linha de contorno
 - | Cor do preenchimento
 - | Espessura da linha
- | Objetos gráficos suportam também:
 - | Moving
 - | Hiding

Objeto GraphWin

- ▮ Representa uma janela na tela onde objetos podem ser desenhados.
- ▮ Possui os seguintes métodos:
 - ▮ **GraphWin(title, width, height)** → constroi uma nova janela gráfica para que se possa desenhar na tela. Os parâmetros são opcionais, os valores default são “Graphics Window,” e 200 x 200.
 - ▮ **plot(x, y, color)** → desenha o pixel (x, y) da janela. A cor é opcional (preto é o valor default).
 - ▮ **setBackground(color)** → preenche a cor de fundo da janela. Cinza é a cor inicial.
 - ▮ **Close()** → fecha a janela.

Objeto GraphWin

- | Possui os seguintes métodos (continuação):
 - | **getMouse()** → pausa até o usuário clicar na janela e retorna o ponto clicado.
 - | **checkMouse()** → semelhante ao `getMouse()`, mas não pausa. Retorna o último ponto clicado na janela.
 - | **checkKey()** → Retorna uma string com o valor da última tecla pressionada (“Left”, “Right”, “space”, etc).
 - | **setCoords(xll, yll, xur, yur)** → configura o sistema de coordenadas da janela. O ponto (xll, yll) indica o canto inferior esquerdo e o ponto (xur, yur) o canto superior direito.

Objeto Graphics

- ‡ Possui as seguintes classes para desenhar objetos:
 - ‡ Point
 - ‡ Line
 - ‡ Circle
 - ‡ Oval
 - ‡ Rectangle
 - ‡ Polygon
 - ‡ Text
- ‡ Todos objetos são iniciados sem preenchimento e com cor de contorno preta.

Objeto Graphics

- | Todos objetos graphics suportam os seguintes métodos:
 - | **setFill(color)** → configura a cor de preenchimento de um objeto.
 - | **setOutline(color)** → configura a cor de contorno de um objeto.
 - | **setWidth(pixels)** → configura a espessura da linha do objeto.
 - | **draw(GraphWin)** → desenha o objeto em um dado GraphWin.
 - | **undraw()** → apaga um objeto de um dado GraphWin.
 - | **move(dx,dy)** → move o objeto dx unidades na direção x e dy unidades na direção y.
 - | **clone()** → duplica um objeto.

Objeto Graphics

- | Método **Point**

- | **Point(x,y)** constrói um ponto dada uma coordenada.
- | **getX()** retorna a coordenada x do ponto.
- | **getY()** retorna a coordenada y do ponto.

- | Método **Line**

- | **Line(point1, point2)** constrói um segmento de linha do ponto 1 ao ponto 2.
- | **getCenter()** retorna um clone do ponto central de um segmento de reta.
- | **getP1(), getP2()** retorna um clone do ponto final correspondente a um segmento.

Objeto Graphics

┆ Método **Circle**

- ┆ **Circle(centerPoint, radius)** constrói um círculo dado um ponto central e um raio.
- ┆ **getCenter()** retorna um clone do ponto central do círculo.
- ┆ **getRadius()** retorna o raio do círculo.
- ┆ **getP1(), getP2()** retorna um clone do ponto correspondente ao canto do círculo.

┆ Método **Rectangle**

- ┆ **Rectangle(point1, point2)** constrói um retângulo dados dois pontos opostos.
- ┆ **getCenter()** retorna um clone do ponto central do retângulo.
- ┆ **getP1(), getP2()** retorna um clone dos pontos usados originalmente para construir o retângulo.

Objeto Graphics

┆ Método **Oval**

- ┆ **Oval(point1, point2)** constrói um objeto oval dentro de uma caixa determinada pelos pontos 1 e 2.
- ┆ **getCenter()** retorna um clone do ponto central do objeto oval.
- ┆ **getP1(), getP2()** retorna um clone dos pontos usados originalmente para construir o objeto oval.

┆ Método **Polygon**

- ┆ **Polygon(point1, point2, point3, ...)** constrói um polígono dados os vértices (pontos).
- ┆ **getPoints()** retorna uma lista contendo clones dos pontos usados para construir o polígono.

Objeto Graphics

▮ Método Text

- ▮ **Text(anchorPoint, string)** constrói um objeto texto com a **string** centralizada no ponto **anchorPoint**.
- ▮ **setText(string)** coloca uma string no objeto.
- ▮ **getText()** retorna a string do objeto.
- ▮ **getAnchor()** retorna um clone do ponto central do objeto.
- ▮ **setFace(family)** muda a fonte ('helvetica', 'courier', 'times roman' e 'arial').
- ▮ **setSize(point)** muda o tamanho da fonte.
- ▮ **setStyle(style)** muda o estilo da fonte ('normal', 'bold', 'italic' e 'bold italic').
- ▮ **setTextColor(color)** muda a cor do texto.

Objeto Graphics

▮ Método **Entry**

- ▮ **Entry(centerPoint, width)** constrói um objeto **entrada** centralizado no ponto **centerPoint** e com largura **width**.
- ▮ **setText(string)** coloca uma string no objeto.
- ▮ **getText()** retorna a string do objeto.
- ▮ **getAnchor()** retorna um clone do ponto central do objeto.
- ▮ **setFace(family)** muda a fonte ('helvetica', 'courier', 'times roman' e 'arial').
- ▮ **setSize(point)** muda o tamanho da fonte.
- ▮ **setStyle(style)** muda o estilo da fonte ('normal', 'bold', 'italic' e 'bold italic').
- ▮ **setTextColor(color)** muda a cor do texto.

Imagens

- ▮ Prove um suporte mínimo a visualização e manipulação de imagens.
- ▮ Suporta os métodos genéricos: **move(dx,dy)**, **draw(graphwin)**, **undraw()**, and **clone()**.
- ▮ Métodos **Image**
 - ▮ **Image(anchorPoint, filename)** constrói uma imagem a partir de um arquivo, centralizada no ponto **anchorPoint**.
 - ▮ **getAnchor()** retorna um clone do ponto central da imagem.
 - ▮ **getWidth()** retorna a largura da imagem.
 - ▮ **getHeight()** retorna a altura da imagem.
 - ▮ **getPixel(x, y)** retorna uma lista [red, green, blue] dos valores RGB do ponto (x,y).
 - ▮ **setPixel(x, y, color)** pinta um ponto (x,y) da imagem com uma cor. Esta é uma operação lenta.
 - ▮ **save(filename)** salva a imagem em um arquivo.

Exemplos

- ▮ Exemplos???
- ▮ Primeiros exercícios???