

Módulo TQSJAN

O módulo `TQSJJan` mostra um objeto tipo `TQSDwg.Dwg` em uma janela Windows, usando um objeto `TQSJJan.Window`.

A apresentação de janelas Windows em Python em geral depende de uma biblioteca GUI de interface de usuário, e existem várias disponíveis. A ligação de um desenho com uma janela depende do tratamento de alguns eventos. Para não tornar o módulo `TQSJJan` dependente de uma biblioteca em particular, os eventos devem ser ligados manualmente à biblioteca escolhida. Isto é mostrado no programa `TSTJan.py`, que lê o desenho `TESTE1.DWG` gerado pelo programa `TSTDwg.py`, e o coloca em uma janela da biblioteca wxPython. Esta biblioteca precisa ser baixada para rodar o exemplo.

A criação de um objeto `TQSJJan.Window` é feito com o uso de dois parâmetros: o *handle* da janela Windows, que é uma variável que caracteriza toda janela no Windows, e um objeto `Dwg`, criado com o módulo `TQSDwg`:

```
import wx

from TQS import TQSJJan, TQSDwg

.....

dwg = TQSDwg.Dwg ()

dwgname = "TESTE1"

if dwg.file.Open (dwgname) != 0:

.....

tqsjan = TQSJJan.Window (hWnd, dwg)
```

Por exemplo, na biblioteca wxPython, uma janela derivada da classe `ws.Window` pode obter este handle pela chamada:

```
hWnd = self.GetHandle ()
```

Métodos e propriedades de TQSJJan

`dwg`

Retorna objeto `Dwg()` associado à esta janela

```
DestroyView ()
```

Destroi e desaloca a janela atual

```
ChangeDwg (dwg)
```

Define novo desenho associado à janela atual

```
orthogonal
```

Modo ortogonal (1) ou normal (0)

```
SetWindow (self, xmin, ymin, xmax, ymax)
```

Define a janela atual (canto esquerdo inferior, canto direito superior).

```
GetWindow ()
```

Retorna xmin, ymin, xmax, ymax da janela atual (canto esquerdo inferior, canto direito superior).

WorldToScreen(x, y)

Converte coordenadas x,y do mundo real para coordenadas de tela, Retorna ix,iy, as coordenadas de tela.

ScreenToWorld (ix, iy)

Converte coordenadas ix,iy de tela para coordenadas do mundo real. Retorna x,y no mundo real.

Regen ()

Regera o desenho na janela Windows.

Window2P (x1, y1, x2, y2)

Redefine a janela atual do desenho por 2 pontos.

WindowScale (scale)

Redefine a janela atual afastando por uma escala.

WindowPan (vx, vy)

Movimenta a janela atual por um vetor.

ZoomTotal ()

Zoom total do desenho atual.

ZoomOut ()

Afasta a visualização por um fator de escala de 0.5.

ZoomPrevious ()

Mostra a janela anterior. Retorna istat != 0 se não há mais janelas

Métodos que interagem com o usuário

Os métodos abaixo esperam que o usuário escolha um ponto na tela e aperte <B1>.

GetPoint ()

Retorna x,y,istat de um ponto. istat != 0 se o usuário apertou <Esc>.

GetPointRubberBand (x1,y1)

Retorna x,y,istat de um ponto, ligando linha elástica com x1,y1. istat != 0 se o usuário apertou <Esc>.

Zoom2P (imod, posxy=[0.,0.])

Pede 2 pontos para uma janela entre eles conforme imod:

imod == (0) 2 pontos (1) cursor + segundo ponto

posxy = [x, y] primeiro ponto para imod == 1

ZoomPan (imod, irub, posxy=[0.,0.])

Faz Pan com 1 ou 2 pontos do usuário conforme imod:

imod(0) Lê do usuário (1) usa posxy

irubTQSJan.RUB_xxx, uma das constantes abaixo

`posxy[ix, iy]` Posição do cursos na janela

`TQSTJan .RUB_LINEAR`

Rubberband linear

`TQSTJan .RUB_RETANG`

Rubberband retangular

`TQSTJan .RUB_NAO`

Sem rubberband

`TQSTJan .RUB_PANDIN`

Caso particular: pan dinamico

Eventos de TQSTJan

`OnPaint(hdc)`

Regera o desenho na janela Windows em um evento `WM_PAINT`. `Hdc` é um "handle para um device contexto", estrutura do Windows.

`OnSize (cx, cy)`

Chamar quando a janela mudar de tamanho, durante evento `WM_SIZE`.

"""

`OnKey (ikeycode, posxy)`

Processa eventos `WM_KEYDOWN` e captura algumas teclas de atalho típicas dos editores gráficos:

`<F8>`Janela por 2 pontos

`<Shift><F8>` Janela total

`<Ctrl><F8>` Janela anterior

`<Alt><F8>` Deslocamento de janela

`<F11>` Zoom out

`<F10>` Liga/desliga ortogonal

`OnMouseMiddle(posxy)`

Trata botão do meio do mouse (rolete pressionado).

`OnWheel(idelta,posxy)`

Processa deslocamentos do rolete do mouse.

Baixando o módulo wxPython

Este programa foi testado com o módulo wxPython baixado e compatível com o Python 3.8. O comando mais comum para baixar e instalar o wxPython é:

`pip install -U wxPython`

Programa TSTJan.py para teste de TQSTJan

Usando o wxPython, o programa tem que ter no mínimo um objeto de aplicação derivado da classe `wxApp`, e um objeto de janela derivado da classe `wx.Frame`. A aplicação principal então será um objeto desta classe:

```

class MainApp (wx.App):
def OnInit(self):
self.m_frame = SampleFrame ()
self.m_frame.Show ()
return True

```

O nosso programa carrega um desenho na memória (o `TESTE1.DWG` gerado pelo `TSTDwg.py`), cria o objeto da aplicação e a janela da classe `SampleFrame`:

```

def main():
dwg = TQSDwg.Dwg ()
dwgname = "TESTE1"
if dwg.file.Open (dwgname) != 0:
return
app = MainApp ()
app.m_frame.m_panel.CreateView (dwg)
app.MainLoop ()

```

Após criar o objeto da aplicação wxPython, é chamada a criação da janela e o loop principal de mensagens. A classe `SampleFrame` criada em `MainApp` gera um container para um painel (uma janela) e um menu, e define uma rotina para tratar o evento de acionamento do menu:

```

class SampleFrame(wx.Frame):
def __init__(self):
super().__init__(parent=None, title='Desenho TQS',
size=(1024, 1024))
self.SetMenuBar (DrawMenuBar())
self.m_panel = DrawingPanel (self)
self.Bind (wx.EVT_MENU, self.m_panel.OnMenu)

```

A rotina `CreateView` efetivamente cria o objeto de janela TQS e liga este objeto ao desenho carregado na memória:

```

def CreateView (self, dwg):
self.m_dwg = dwg
self.m_tqsjan = TQSJan.Window (self.m_hwnd, self.m_dwg)

```

A classe `DrawingPanel` define as características da janela (tamanho 1024 x 1024) e as rotinas de tratamento de eventos:

```

class DrawingPanel(wx.Window):
def __init__(self, parent):
super().__init__(parent, -1)
self.SetSize(1024, 1024)

```

```

self.SetBackgroundColour(wx.Colour(0, 0, 0))

self.m_hwnd = self.GetHandle()

self.Bind (wx.EVT_PAINT, self.OnPaint)

self.Bind (wx.EVT_KEY_DOWN, self.KeyDown)

self.Bind (wx.EVT_MOUSEWHEEL, self.OnWheel)

self.Bind (wx.EVT_MOUSEWHEEL, self.OnWheel)

self.Bind (wx.EVT_MIDDLE_DOWN , self.OnMouseMiddle)

```

As demais rotinas recebem os eventos declarados e chamam as funções correspondentes em [TQSTJan](#).

O que faz o programa TSTJan.py

Apesar de pequeno, o programa [TSTJan.py](#):

Permite a visualização do desenho TESTE1.DWG.

Permite operações de janela com teclas de atalho – Janela por 2 pontos (<F8>), Janela total (<Shift><F8>), Janela anterior (<Ctrl><F8>), Deslocamento de janela (<Alt><F8>), Zoom out (<F11>) e controle de modo ortogonal (<F10>).

Aceita controle de janela seja por movimento do rolete do mouse (zoom in e out) ou por movimento com o botão do rolete apertado.

Define um menu *dropdown* com os comandos de janela acionados por menu, mais um comando para desenhar linhas.

