

Help on module elvis:

NAME

elvis - # -*- coding: utf-8 -*-

FILE

../elvis.py

FUNCTIONS

QT_TRANSLATE_NOOP(...)

QT_TR_NOOP(...)

Q_ARG(...)

Q_ENUMS(...)

Q_FLAGS(...)

Q_RETURN_ARG(...)

SIGNAL(...)

SLOT(...)

bin(...)

bom(...)

center(...)

dec(...)

endl(...)

fixed(...)

flush(...)

forcepoint(...)

forcesign(...)

hex(...)

left(...)

lowercasebase(...)

lowercasedigits(...)

noforcepoint(...)

noforcesign(...)

noshowbase(...)

oct(...)

pyqtRemoveInputHook(...)

pyqtRestoreInputHook(...)

pyqtSignature(...)

qAbs(...)

qAddPostRoutine(...)

qAlpha(...)

qBlue(...)

qChecksum(...)

qCompress(...)

qCritical(...)

qDebug(...)

qDrawPlainRect(...)

qDrawShadeLine(...)

qDrawShadePanel(...)

qDrawShadeRect(...)

qDrawWinButton(...)

qDrawWinPanel(...)

qErrnoWarning(...)

qFatal(...)

qFuzzyCompare(...)

qGray(...)

qGreen(...)

qInf(...)

qInstallMsgHandler(...)

qIsFinite(...)

qIsGray(...)

qIsInf(...)

qIsNaN(...)

qIsNull(...)

qQNaN(...)

qRed(...)

qRegisterResourceData(...)

qRemovePostRoutine(...)

qRgb(...)

qRgba(...)

qRound(...)

qRound64(...)

qSNaN(...)

qSetFieldWidth(...)

qSetPadChar(...)

qSetRealNumberPrecision(...)

qSharedBuild(...)

qSwap(...)

qUncompress(...)

qUnregisterResourceData(...)

qVersion(...)

qWarning(...)

qrand(...)

qsrand(...)

qstrcmp(...)
qstrcpy(...)
qstrdup(...)
qstricmp(...)
qstrlen(...)
qstrncmp(...)
qstrncpy(...)
qstrnicmp(...)
qstrnlen(...)
qt_set_sequence_auto_mnemonic(...)
qt_x11_wait_for_window_manager(...)
reset(...)
right(...)
scientific(...)
showbase(...)
uppercasebase(...)
uppercasedigits(...)
ws(...)

DATA

PYQT_VERSION = 263172
PYQT_VERSION_STR = '4.4.4'
QT_VERSION = 263424
QT_VERSION_STR = '4.5.0'
QtCriticalMsg = 2
QtDebugMsg = 0
QtFatalMsg = 3
QtSystemMsg = 2
QtWarningMsg = 1
__autor__ = 'Le\x3\xb4nidas S. Barbosa (kirotawa)'
__version__ = 0.0
app = <PyQt4.QtGui.QApplication object at 0x8a57c6c>
form = <main.mainwindow.MainWindow object at 0x8a57cac>
qApp = <PyQt4.QtGui.QApplication object at 0xb7d35d6c>

VERSION
0.0

Help on module formas:

NAME
formas - # -*- coding: utf-8 -*-

FILE
../formas.py

CLASSES

- __builtin__.object
- FormasGeometricas
- SquareControl
- Vertice
- Point

```
class FormasGeometricas(__builtin__.object)
| Classe base para todas as formas (linha, quadrado, circulo, elipse e curva)
|
| Methods defined here:
|
| Colide(self, mouse)
|     Método usado para detectar colisões, ou seja, verificar se uma forma foi selecionada
|
| Rotate(self, degree, point)
|     Método para aplicar rotação as formas
|
| Scale(self, zoom, typ)
|     Método usado para escalar a forma
|
| Translate(self, desloc, direc)
|     Método usado para transladar a forma
|
| __init__(self, id)
|     Construtor da classe, que inicializa as cores de borda e de preenchimento para o default
|
| addPoint(self, ponto)
|     Adiciona pontos a forma
|
| drawControlPoints(self)
|     Desenha os pontos de controle da forma
|
| drawPoint(self, ponto)
|     Desenha os pontos de acordo com sua posição e cor
|
| drawPointFill(self, ponto)
|     Metodo usado para desenhar pontos com cor no preenchimento de formas
|
| drawShape(self)
```

Método abstrato para as classes filhas

`drawVertice(self, x, y)`

Desenha os vertices dos pontos de controle

`move(self, p1, p2)`

Método usado para mover a forma de lugar de forma interativa

Data descriptors defined here:

`__dict__`

dictionary for instance variables (if defined)

`__weakref__`

list of weak references to the object (if defined)

`class Point(__builtin__.object)`

Classe que define pontos ou vertices

Methods defined here:

`__init__(self, ponto)`

Data descriptors defined here:

`__dict__`

dictionary for instance variables (if defined)

`__weakref__`

list of weak references to the object (if defined)

`class SquareControl(FormasGeometricas)`

Classe que desenha um quadrado/retangulo

Method resolution order:

`SquareControl`

`FormasGeometricas`

`__builtin__.object`

Methods defined here:

`Bresenham(self, p1, p2)`

Método de Bresenham para desenharas quatro linhas do quadrado dos pontos de controle

`__init__(self, id)`

`drawShape(self)`

Methods inherited from FormasGeometricas:

Colide(self, mouse)

Método usado para detectar colisões, ou seja, verificar se uma forma foi selecionada

Rotate(self, degree, point)

Método para aplicar rotação as formas

Scale(self, zoom, typ)

Método usado para escalar a forma

Translate(self, desloc, direc)

Método usado para transladar a forma

addPoint(self, ponto)

Adiciona pontos a forma

drawControlPoints(self)

Desenha os pontos de controle da forma

drawPoint(self, ponto)

Desenha os pontos de acordo com sua posição e cor

drawPointFill(self, ponto)

Metodo usado para desenhar pontos com cor no preenchimento de formas

drawVertice(self, x, y)

Desenha os vertices dos pontos de controle

move(self, p1, p2)

Método usado para mover a forma de lugar de forma interativa

Data descriptors inherited from FormasGeometricas:

__dict__

dictionary for instance variables (if defined)

__weakref__

list of weak references to the object (if defined)

class Vertice(FormasGeometricas)

Classe que define o vertice

Method resolution order:

Vertice

FormasGeometricas

__builtin__.object

Methods defined here:

__init__(self, id)

| drawShape(self)

| -----
| Methods inherited from FormasGeometricas:

| Colide(self, mouse)

| Método usado para detectar colisões, ou seja, verificar se uma forma foi selecionada

| Rotate(self, degree, point)

| Método para aplicar rotação as formas

| Scale(self, zoom, typ)

| Método usado para escalar a forma

| Translate(self, desloc, direc)

| Método usado para transladar a forma

| addPoint(self, ponto)

| Adiciona pontos a forma

| drawControlPoints(self)

| Desenha os pontos de controle da forma

| drawPoint(self, ponto)

| Desenha os pontos de acordo com sua posição e cor

| drawPointFill(self, ponto)

| Metodo usado para desenhar pontos com cor no preenchimento de formas

| drawVertice(self, x, y)

| Desenha os vertices dos pontos de controle

| move(self, p1, p2)

| Método usado para mover a forma de lugar de forma interativa

| -----
| Data descriptors inherited from FormasGeometricas:

| dict

| dictionary for instance variables (if defined)

| weakref

| list of weak references to the object (if defined)

FUNCTIONS

cos(...)

cos(x)

Return the cosine of x (measured in radians).

sin(...)

sin(x)

Return the sine of x (measured in radians).

```
sqrt(...)
sqrt(x)
```

Return the square root of x.

Help on module circle:

NAME

circle - # -*- coding: utf-8 -*-

FILE

../circle.py

CLASSES

```
formas.FormasGeometricas(__builtin__.object)
    Circle
```

```
class Circle(formas.FormasGeometricas)
```

```
|   Classe que desenha um circulo
```

```
|
```

```
|   Method resolution order:
```

```
|       Circle
```

```
|       formas.FormasGeometricas
```

```
|       __builtin__.object
```

```
|
```

```
|   Methods defined here:
```

```
|
```

```
|   BresenhamCircle(self, cx, cy, radius)
```

```
|       Método que usa o algoritmo de Bresenham para desenharm um circulo ponto a ponto
```

```
|
```

```
|   __init__(self, id)
```

```
|
```

```
|   addRadius(self)
```

```
|
```

```
|   drawShape(self)
```

```
|
```

```
|   floodFill(self, radius, cx, cy)
```

```
|
```

```
|   plot4points(self, cx, cy, x, y)
```

```
|       Função auxiliar a função plot8points para desenharm o algoritmo de Bresenham
```

```
|
```

```
|   plot8points(self, cx, cy, x, y)
```

```
|       Função auxiliar para desenharm o circulo de Bresenham
```

```
|
```

```
|   -----
```

```
|   Methods inherited from formas.FormasGeometricas:
```

```
|
```

Colide(self, mouse)	Método usado para detectar colisões, ou seja, verificar se uma forma foi selecionada
Rotate(self, degree, point)	Método para aplicar rotação as formas
Scale(self, zoom, typ)	Método usado para escalar a forma
Translate(self, desloc, direc)	Método usado para transladar a forma
addPoint(self, ponto)	Adiciona pontos a forma
drawControlPoints(self)	Desenha os pontos de controle da forma
drawPoint(self, ponto)	Desenha os pontos de acordo com sua posição e cor
drawPointFill(self, ponto)	Metodo usado para desenhar pontos com cor no preenchimento de formas
drawVertice(self, x, y)	Desenha os vertices dos pontos de controle
move(self, p1, p2)	Método usado para mover a forma de lugar de forma interativa
Data descriptors inherited from formas.FormasGeometricas:	
__dict__	dictionary for instance variables (if defined)
__weakref__	list of weak references to the object (if defined)

FUNCTIONS

cos(...)
cos(x)

Return the cosine of x (measured in radians).

sin(...)
sin(x)

Return the sine of x (measured in radians).

sqrt(...)
sqrt(x)

Return the square root of x.

Help on module ellipse:

NAME

ellipse - #-*- coding: utf-8-#-

FILE

../ellipse.py

CLASSES

formas.FormasGeometricas(__builtin__.object)

Ellipse

class Ellipse(formas.FormasGeometricas)

| Classe que desenha e controla a ellipse

| Method resolution order:

| Ellipse

| formas.FormasGeometricas

| __builtin__.object

| Methods defined here:

| __init__(self, id)

| drawEllipse(self, a, b, cx, cy)

| Método interativo para desenhar uma ellipse ponto a ponto

| drawPointEllipse(self, x, y, cx, cy)

| drawShape(self)

| floodFill(self, a, b, cx, cy)

| Método de preenchimento interativo

| Methods inherited from formas.FormasGeometricas:

| Colide(self, mouse)

| Método usado para detectar colisões, ou seja, verificar se uma forma foi selecionada

| Rotate(self, degree, point)

| Método para aplicar rotação as formas

| Scale(self, zoom, typ)

| Método usado para escalar a forma

| Translate(self, desloc, direc)

| Método usado para transladar a forma

	<code>addPoint(self, ponto)</code>	
		Adiciona pontos a forma
	<code>drawControlPoints(self)</code>	
		Desenha os pontos de controle da forma
	<code>drawPoint(self, ponto)</code>	
		Desenha os pontos de acordo com sua posição e cor
	<code>drawPointFill(self, ponto)</code>	
		Metodo usado para desenhar pontos com cor no preenchimento de formas
	<code>drawVertice(self, x, y)</code>	
		Desenha os vertices dos pontos de controle
	<code>move(self, p1, p2)</code>	
		Método usado para mover a forma de lugar de forma interativa

	Data descriptors inherited from formas.FormasGeometricas:	
	<code>dict</code>	
		dictionary for instance variables (if defined)
	<code>weakref</code>	
		list of weak references to the object (if defined)

FUNCTIONS

`cos(...)`

`cos(x)`

Return the cosine of x (measured in radians).

`fabs(...)`

`fabs(x)`

Return the absolute value of the float x.

`sin(...)`

`sin(x)`

Return the sine of x (measured in radians).

`sqrt(...)`

`sqrt(x)`

Return the square root of x.

Help on module square:

NAME

square - # -*- coding: utf-8 -*-

FILE

../square.py

CLASSES

formas.FormasGeometricas(__builtin__.object)
Square

class Square(formas.FormasGeometricas)

| Classe que desenha um quadrado/retangulo

| Method resolution order:

| Square

| formas.FormasGeometricas

| __builtin__.object

| Methods defined here:

| Bresenham(self, p1, p2)

| Algoritmo de bresenham para desenhar linhas

| __init__(self, id)

| drawShape(self)

| Desenha o retangulo

| - passados os 4 pontos do retangulo este metodo o desenha atraves do algoritmo de linhas de

Bresenham

| - se a variavel de controle fulfilled for verdadeira, desenha a forma com preenchimento

| fillFlood(self, x, y, xmin, xmax, ymin, ymax)

| Método de preenchimento de formas Flood Fill

| - Dadas os pontos de min e max dos eixos na forma este metodo preenche tudo dentro da

forma com pontos da cor estabelecida

| -----
| Methods inherited from formas.FormasGeometricas:

| Colide(self, mouse)

| Método usado para detectar colisões, ou seja, verificar se uma forma foi selecionada

| Rotate(self, degree, point)

| Método para aplicar rotação as formas

| Scale(self, zoom, typ)

| Método usado para escalar a forma

| Translate(self, desloc, direc)

| Método usado para transladar a forma

| addPoint(self, ponto)

| Adiciona pontos a forma

	<code>drawControlPoints(self)</code>	
		Desenha os pontos de controle da forma
	<code>drawPoint(self, ponto)</code>	
		Desenha os pontos de acordo com sua posição e cor
	<code>drawPointFill(self, ponto)</code>	
		Metodo usado para desenhar pontos com cor no preenchimento de formas
	<code>drawVertice(self, x, y)</code>	
		Desenha os vertices dos pontos de controle
	<code>move(self, p1, p2)</code>	
		Método usado para mover a forma de lugar de forma interativa

	Data descriptors inherited from formas.FormasGeometricas:	
	<code>__dict__</code>	
		dictionary for instance variables (if defined)
	<code>__weakref__</code>	
		list of weak references to the object (if defined)

FUNCTIONS

`cos(...)`

`cos(x)`

Return the cosine of x (measured in radians).

`sin(...)`

`sin(x)`

Return the sine of x (measured in radians).

`sqrt(...)`

`sqrt(x)`

Return the square root of x.

Help on module line:

NAME

line - # -*- coding: utf-8 -*-

FILE

../line.py

CLASSES

formas.FormasGeometricas(`__builtin__.object`)

Line

class Line(formas.FormasGeometricas)

Classe que desenha a linha Bresenham

Method resolution order:

Line

formas.FormasGeometricas

__builtin__.object

Methods defined here:

Bresenham(self, p1, p2)

Algoritmo de bresenham para desenhar linhas

__init__(self, id)

drawShape(self)

Methods inherited from formas.FormasGeometricas:

Colide(self, mouse)

Método usado para detectar colisões, ou seja, verificar se uma forma foi selecionada

Rotate(self, degree, point)

Método para aplicar rotação as formas

Scale(self, zoom, typ)

Método usado para escalar a forma

Translate(self, desloc, direc)

Método usado para transladar a forma

addPoint(self, ponto)

Adiciona pontos a forma

drawControlPoints(self)

Desenha os pontos de controle da forma

drawPoint(self, ponto)

Desenha os pontos de acordo com sua posição e cor

drawPointFill(self, ponto)

Metodo usado para desenhar pontos com cor no preenchimento de formas

drawVertice(self, x, y)

Desenha os vertices dos pontos de controle

move(self, p1, p2)

Método usado para mover a forma de lugar de forma interativa

Data descriptors inherited from formas.FormasGeometricas:

`__dict__`
dictionary for instance variables (if defined)

`__weakref__`
list of weak references to the object (if defined)

Help on module glwidget:

NAME

glwidget - #-*- coding: utf-8-*

FILE

../glwidget.py

CLASSES

PyQt4.QtOpenGL.QGLWidget(PyQt4.QtGui.QWidget)
 GIWidget

class GIWidget(PyQt4.QtOpenGL.QGLWidget)

 Painel OpenGL

 Method resolution order:

 GIWidget
 PyQt4.QtOpenGL.QGLWidget
 PyQt4.QtGui.QWidget
 PyQt4.QtCore.QObject
 PyQt4.QtGui.QPaintDevice
 sip.wrapper
 __builtin__.object

 Methods defined here:

 ExportSVG(self)

 Metodo acionado para exportar imagem .svg

 - Salva um arquivo XML com as características de um arquivo .SVG dado um nome e uma localização pelo usuário

 RGBtoHex(self, color)

 Metodo que converte as cores de RGB float para RGB em hexadecimal

 __init__(self, parent=None)

 Metodo construtor para inicializar o que for preciso

 circuloControl(self)

 Metodo de controle do circulo e da elipse

 colorControl(self)

 Controle de cores de borda

| colorfillControl(self)

| Metodo que define a cor e manda preencher os objetos

| copyControl(self)

| Metodo que controla a copia de objetos

| drawGrid(self, size)

| Desenha um grid dadas dimensões passadas pelo usuário

| grade(self)

| Chamada para definição da grade

| - É dado ao usuário o direito de escolher as dimensões da grade.

| - Se uma grade já está desenhada ele retira a grade, senão ele a desenha com as dimensões

passadas

| initializeGL(self)

| Metodo do glwidget para inicializar o opengl

| - glClearColor(1.0,1.0,1.0,1.0): Limpa o buffer para branco.

| initializeVars(self)

| Metodo para iniciar alguns valores de variaveis de controle interno

| keyPressEvent(self, event)

| Controla eventos de pressionamento de teclas

| - Controla o pan da tela realizando translações nas formas de acordo com a sua orientação

dada pelo teclado de setas.

| - Controla se uma elipse deve ser desenhada ao inves de um circulo

| - Apaga uma forma se esta estiver selecionada e a telca delete for acionada

| keyReleaseEvent(self, event)

| Controla os eventos da liberação das teclas do teclado

| lineControl(self)

| Método de controle de linhas

| mouseMoveEvent(self, event)

| Metodo responsável principalmente por capturar o segundo ponto de cada forma

| mousePressEvent(self, event)

| Método de controle de eventos de de pressionamento do mouse

| - Neste metodo são tratado os cliques que irão dar origem a objetos, controlar a seleção, a

copia, a rotação.

| - Nesse metodo é capturado o primeiro ponto de cada forma.

| mouseReleaseEvent(self, event)

| Metodo analogo ao press, controla o release do mouse e desenha as formas de fato

| newFile(self)

| Metodo chamado quando um novo arquivo é acionado

| - Pergunta se é desejavel salvar o arquivo atual

| - Limpa a lista de formas e inicializa as variaveis de controle

openElvisfile(self)

Método mágico que abre arquivo .elv

- Todos os objetos salvos no arquivo .elv são recarregados para lista de formas

paintGL(self)

Método responsável por pintar objetos no painel opengl

- Controla o zoom, o grid, o desenho o desenho das formas.

```
if self.zoomin_ == True:
```

```
    self.zoomin_ = False
```

```
    for i in self.shapes:
```

```
        i.Scale(0.75,'in')
```

```
if self.zoomout_ == True:
```

```
    self.zoomout_ = False
```

```
    for i in self.shapes:
```

```
        i.Scale(0.75,'out')
```

```
if self.grid:
```

```
    self.drawGrid(self.gridsize)
```

```
for i in self.shapes:
```

```
    i.drawShape()
```

```
    if self.pcontrole or i.selected:
```

```
        i.drawControlPoints()
```

```
glPopMatrix()
```

```
glFlush()
```

pontosControle(self)

Método que controla a inserção e remoção de pontos de controle

resizeGL(self, width, height)

Método que resetar as dimensões da tela do opengl

rotationControl(self)

Método que controla a rotação dos objetos

saveElvisfile(self)

Método mágico que salva as imagens em .elv

- Todos os objetos/formas são exportados para um arquivo

select(self)

Método de controle da seleção

squareControl(self)

Método de controle do retângulo/quadrado

vertice(self)

Método de controle de inserção de vértices

zoomin(self)

Método que efetua o zoom in

```
| zoomout(self)
| Metodo que efetua o zoom out
|
```

Help on module mainwindow:

NAME

mainwindow - # -*- coding: utf-8 -*-

FILE

../mainwindow.py

CLASSES

```
PyQt4.QtGui.QMainWindow(PyQt4.QtGui.QWidget)
    MainWindow(PyQt4.QtGui.QMainWindow, ui_mainwindow.Ui_MainWindow)
ui_mainwindow.Ui_MainWindow(__builtin__.object)
    MainWindow(PyQt4.QtGui.QMainWindow, ui_mainwindow.Ui_MainWindow)
```

```
class MainWindow(PyQt4.QtGui.QMainWindow, ui_mainwindow.Ui_MainWindow)
```

```
| Classe responsável pelos widgets na tela (menu, botões, etc)
```

```
| Method resolution order:
```

```
|   MainWindow
|   PyQt4.QtGui.QMainWindow
|   PyQt4.QtGui.QWidget
|   PyQt4.QtCore.QObject
|   PyQt4.QtGui.QPaintDevice
|   sip.wrapper
|   ui_mainwindow.Ui_MainWindow
|   __builtin__.object
```

```
| Methods defined here:
```

```
|   __init__(self, parent=None)
|   Construtor
```

```
|   about(self)
|   Metodo que chama a janela about
```

```
|   makeActionMenu(self)
|   Cria as acoes do menu atraves das funcoes que eles executam
```

```
|   makeActions(self)
|   Cria os botoes os adiciona ao toolBar
```

```
|   makeMenu(self)
|   Cria os rotulos dos menus no menuBar
```

```
|   on_actionQuit_triggered(self)
|   Chamado automaticamente quando o evento de sair eh chamado
```