

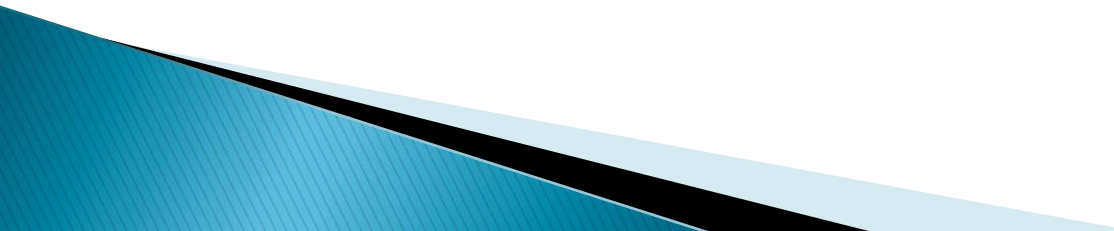
Programação Gráfica

Professor Rodrigo Piva

OpenGL Intermediário



Agenda

- 1 OpenGL
 - 2 Programa Avançado
 - 3 Comentando o Código
 - 4 Exercício
- 

Programa – Código (MAIN)

```
if __name__ == '__main__':  
    pg.init()    display = (400, 400)  
    pg.display.set_mode(display, DOUBLEBUF|OPENGL | pg.OPENGLBLIT)  
    gluPerspective(45, (display[0]/display[1]), 0.1, 50.0)  
    glTranslatef(0.0, 0.0, -5)  
    while True:  
        for event in pg.event.get():  
            if event.type == pg.QUIT:  
                pg.quit()  
                quit()  
            glRotatef(1, 1, 1, 1)  
            glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)  
            ladoum() #branco  
            ladodois() #roxo  
            ladotres() #multicor  
            ladoquatro() #verde  
            ladocinco() #azul  
            ladoseis() #vermelho  
            plano()  
            #wireCube()  
            pg.display.flip()  
            pg.time.wait(50)
```

Programa – Código (MAIN)

```
if __name__ == '__main__':  
    pg.init()    display = (400, 400)  
    pg.display.set_mode(display, DOUBLEBUF|OPENGL | pg.OPENGLBLIT)  
    gluPerspective(45, (display[0]/display[1]), 0.1, 50.0)  
    glTranslatef(0.0, 0.0, -5)  
    while True:  
        for event in pg.event.get():  
            if event.type == pg.QUIT:  
                pg.quit()  
                quit()  
            glRotatef(1, 1, 1, 1)  
            glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)  
            ladoum() #branco  
            ladodois() #roxo  
            ladotres() #multicor  
            ladoquatro() #verde  
            ladocinco() #azul  
            ladoseis() #vermelho  
            plano()  
            #wireCube()  
            pg.display.flip()  
            pg.time.wait(50)
```

Programa – Código (MAIN)

def inicia():

```
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    glFrustum(-1, 1, -0.75, 0.75, 1, 40)
    #glOrtho(0, 250, 0, 250, -1, 1)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()
    glTranslatef(0, 0, -2)
    glRotatef(50, 1, 0, 0)
    glRotatef(50, 0, 1, 0)
```

def plano():

```
    glColor3f(1.0, 1.0, 1.0)
    glLineWidth(4)
    glBegin(GL_LINES)
    for i in range(-4,4,1):
        glVertex3f(i, -1.5, 4)
        glVertex3f(i, -1.5, -4)
        glVertex3f(4, -1.5, i)
        glVertex3f(-4, -1.5, i)
    glEnd()
    glFlush()
```

Programa – Código

def inicia():

glMatrixMode(GL_PROJECTION)

glLoadIdentity()

glFrustum(-1, 1, -0.75, 0.75, 1, 40)

#glOrtho(0, 250, 0, 250, -1, 1)

glMatrixMode(GL_MODELVIEW)

glLoadIdentity()

glTranslatef(0, 0, -2)

glRotatef(50, 1, 0, 0)

glRotatef(50, 0, 1, 0)

----- INICIA O MODEL EM PROJEÇÃO

-----CARREGA A IDENTIDADE

-----CRIA O MODELO DE PROJEÇÃO ORTOGONAL

-----MODELO DE PROJEÇÃO ORTOGONAL 2D

-----CARREGA O MODELO DE VIZUALIZAÇÃO

-----CARREGA A IDENTIDADE

-----FAZ A TRANSLAÇÃO PARA Z= -2

----- FAZ A ROTAÇÃO DE 50 GRAUS NO EIXO X

----- FAZ A ROTAÇÃO DE 50 GRAUS NO EIXO Y

def plano():

glColor3f(1.0, 1.0, 1.0)

glLineWidth(4)

glBegin(GL_LINES)

for i in range(-4,4,1):

glVertex3f(i, -1.5, 4)

glVertex3f(i, -1.5, -4)

glVertex3f(4, -1.5, i)

glVertex3f(-4, -1.5, i)

glEnd()

glFlush()



Programa – Código

def inicia():

glMatrixMode(GL_PROJECTION)

glLoadIdentity()

glFrustum(-1, 1, -0.75, 0.75, 1, 40)

#glOrtho(0, 250, 0, 250, -1, 1)

glMatrixMode(GL_MODELVIEW)

glLoadIdentity()

glTranslatef(0, 0, -2)

glRotatef(50, 1, 0, 0)

glRotatef(50, 0, 1, 0)

----- INICIA O MODEL EM PROJEÇÃO

-----CARREGA A IDENTIDADE

-----CRIAR O MODELO DE PROJEÇÃO ORTOGONAL

-----MODELO DE PROJEÇÃO ORTOGONAL 2D

-----CARREGA O MODELO DE VIZUALIZAÇÃO

-----CARREGA A IDENTIDADE

-----FAZ A TRANSLAÇÃO PARA Z= -2

----- FAZ A ROTAÇÃO DE 50 GRAUS NO EIXO X

----- FAZ A ROTAÇÃO DE 50 GRAUS NO EIXO Y

def plano():

glColor3f(1.0, 1.0, 1.0)

glLineWidth(4)

glBegin(GL_LINES)

for i in range(-4,4,1):

glVertex3f(i, -1.5, 4)

glVertex3f(i, -1.5, -4)

glVertex3f(4, -1.5, i)

glVertex3f(-4, -1.5, i)

glEnd()

glFlush()

----- DEFINE A COR COMO BRANCO

----- DEFINE A LARGURA DA LINHA EM 4 PIXELS

----- INICIALIZA O GL EM MODO LINHA

----- CRIA A GRADE DO PLANO COMO VERTICES

-----FINALIZA O DESENHO

-----LIMPA O BUFFER DO OPENG

Programa – Código (MAIN)

```
def apaga():  
    #glClearColor(0.1, 0.39, 0.88, 1.0)  
    glClearColor(0.0, 0.0, 0.0, 0.0)  
    glClear(GL_COLOR_BUFFER_BIT)
```

```
def rotaciona():  
    glRotatef(5, 0,0, 10)  
    glFlush()
```

Programa – Código (MAIN)

```
def apaga():
```

```
#glClearColor(0.1, 0.39, 0.88, 1.0)
```

```
glClearColor(0.0, 0.0, 0.0, 0.0)
```

----- DEFINE A COR COMO PRETO

```
glClear(GL_COLOR_BUFFER_BIT)
```

----- LIMPA O BUFFER

```
def rotaciona():
```

```
glRotatef(5, 0,0, 10)
```

----- ROTACIONA 5 GRAUS NO EIXO Z

glFlush()

```
----- LIMPA O BUFFER DE ESCRITA
```

Programa – Código (MAIN)

```
def ladoum():  
    glBegin(GL_POLYGON)  
    glColor3f( 1.0, 1.0, 1.0 )  
    glVertex3f( 0.5, -0.5, 0.5 )  
    glVertex3f( 0.5, 0.5, 0.5 )  
    glVertex3f( -0.5, 0.5, 0.5 )  
    glVertex3f( -0.5, -0.5, 0.5 )  
    glEnd()  
    glFlush()
```

```
def ladodois():  
    glBegin(GL_POLYGON)  
    glColor3f( 1.0, 0.0, 1.0 )  
    glVertex3f( 0.5, -0.5, -0.5 )  
    glVertex3f( 0.5, 0.5, -0.5 )  
    glVertex3f( 0.5, 0.5, 0.5 )  
    glVertex3f( 0.5, -0.5, 0.5 )  
    glEnd()  
    glFlush()
```

Programa – Código (MAIN)

def ladoum():

glBegin(GL_POLYGON)

glColor3f(1.0, 1.0, 1.0)

glVertex3f(0.5, -0.5, 0.5)

glVertex3f(0.5, 0.5, 0.5)

glVertex3f(-0.5, 0.5, 0.5)

glVertex3f(-0.5, -0.5, 0.5)

glEnd()

glFlush()

-

----- INICIA O OPENGL EM MODO POLIGONO

----- DEFINE A COR DE ESCRITA BRANCA

----- CRIA UM VERTICE NO PONTO DET.

----- CRIA UM VERTICE NO PONTO DET.

----- CRIA UM VERTICE NO PONTO DET.

----- CRIA UM VERTICE NO PONTO DET.

----- FINALIZA A ESCRITA

----- LIMPA O BUFFER

def ladodois():

glBegin(GL_POLYGON)

glColor3f(1.0, 0.0, 1.0)

glVertex3f(0.5, -0.5, -0.5)

glVertex3f(0.5, 0.5, -0.5)

glVertex3f(0.5, 0.5, 0.5)

glVertex3f(0.5, -0.5, 0.5)

glEnd()

glFlush()

-

----- INICIA O OPENGL EM MODO POLIGONO

----- DEFINE A COR DE ESCRITA ROXO

----- CRIA UM VERTICE NO PONTO DET.

----- CRIA UM VERTICE NO PONTO DET.

----- CRIA UM VERTICE NO PONTO DET.

----- CRIA UM VERTICE NO PONTO DET.

----- FINALIZA A ESCRITA

----- LIMPA O BUFFER

Programa – Código (MAIN)

```
def ladotres():  
    glBegin(GL_POLYGON);  
    glColor3f( 1.0, 0.0, 0.0 )  
    glVertex3f( 0.5, -0.5, -0.5 )  
    glColor3f( 0.0, 1.0, 0.0 )  
    glVertex3f( 0.5, 0.5, -0.5 )  
    glColor3f( 0.0, 0.0, 1.0 )  
    glVertex3f( -0.5, 0.5, -0.5 )  
    glColor3f( 1.0, 0.0, 1.0 )  
    glVertex3f( -0.5, -0.5, -0.5 )  
    glEnd()  
    glFlush()
```

```
def ladoquatro():  
    glBegin(GL_POLYGON)  
    glColor3f( 0.0, 1.0, 0.0 )  
    glVertex3f( -0.5, -0.5, 0.5 )  
    glVertex3f( -0.5, 0.5, 0.5 )  
    glVertex3f( -0.5, 0.5, -0.5 )  
    glVertex3f( -0.5, -0.5, -0.5 )  
    glEnd()  
    glFlush()
```

Programa – Código (MAIN)

```
def ladotres():  
    glBegin(GL_POLYGON);  
    glColor3f( 1.0, 0.0, 0.0 )  
    glVertex3f( 0.5, -0.5, -0.5 )  
    glColor3f( 0.0, 1.0, 0.0 )  
    glVertex3f( 0.5, 0.5, -0.5 )  
    glColor3f( 0.0, 0.0, 1.0 )  
    glVertex3f( -0.5, 0.5, -0.5 )  
    glColor3f( 1.0, 0.0, 1.0 )  
    glVertex3f( -0.5, -0.5, -0.5 )  
    glEnd()  
    glFlush()  
  
def ladoquatro():  
    glBegin(GL_POLYGON)  
    glColor3f( 0.0, 1.0, 0.0 )  
    glVertex3f( -0.5, -0.5, 0.5 )  
    glVertex3f( -0.5, 0.5, 0.5 )  
    glVertex3f( -0.5, 0.5, -0.5 )  
    glVertex3f( -0.5, -0.5, -0.5 )  
    glEnd()  
    glFlush()
```

-
----- INICIA O OPENGL EM MODO POLIGONO
----- DEFINE A COR DE ESCRITA VERMELHO
----- CRIA UM VERTICE NO PONTO DET.
----- DEFINE A COR DE ESCRITA VERDE
----- CRIA UM VERTICE NO PONTO DET.
----- DEFINE A COR DE ESCRITA AZUL
----- CRIA UM VERTICE NO PONTO DET.
----- DEFINE A COR DE ESCRITA ROXO
----- CRIA UM VERTICE NO PONTO DET.
----- FINALIZA A ESCRITA
----- LIMPA O BUFFER

----- INICIA O OPENGL EM MODO POLIGONO
----- DEFINE A COR DE ESCRITA VERDE
----- CRIA UM VERTICE NO PONTO DET.
----- CRIA UM VERTICE NO PONTO DET.
----- CRIA UM VERTICE NO PONTO DET.
----- CRIA UM VERTICE NO PONTO DET.
----- FINALIZA A ESCRITA
----- LIMPA O BUFFER

Programa – Código (MAIN)

```
def ladocinco():
```

```
    glBegin(GL_POLYGON)
    glColor3f( 0.0, 0.0, 1.0 )
    glVertex3f( 0.5, 0.5, 0.5 )
    glVertex3f( 0.5, 0.5, -0.5 )
    glVertex3f( -0.5, 0.5, -0.5 )
    glVertex3f( -0.5, 0.5, 0.5 )
    glEnd()
    glFlush()
```

```
def ladoseis():
```

```
    glBegin(GL_POLYGON)
    glColor3f( 1.0, 0.0, 0.0 )
    glVertex3f( 0.5, 0.5, 0.5 ) #trasdir
    glVertex3f( 0.5, -0.5, 0.5 ) #diandir
    glVertex3f( -0.5, -0.5, 0.5 ) #dianesq
    glVertex3f( -0.5, 0.5, 0.5 ) #trsesq
    glEnd()
    glFlush()
```

Programa – Código (MAIN)

def ladocinco():

glBegin(GL_POLYGON)

glColor3f(0.0, 0.0, 1.0)

glVertex3f(0.5, 0.5, 0.5)

glVertex3f(0.5, 0.5, -0.5)

glVertex3f(-0.5, 0.5, -0.5)

glVertex3f(-0.5, 0.5, 0.5)

glEnd()

glFlush()

-

----- INICIA O OPENGL EM MODO POLIGONO

----- DEFINE A COR DE ESCRITA AZUL

----- CRIA UM VERTICE NO PONTO DET.

----- CRIA UM VERTICE NO PONTO DET.

----- CRIA UM VERTICE NO PONTO DET.

----- CRIA UM VERTICE NO PONTO DET.

----- FINALIZA A ESCRITA

----- LIMPA O BUFFER

def ladoseis():

glBegin(GL_POLYGON)

glColor3f(1.0, 0.0, 0.0)

glVertex3f(0.5, 0.5, 0.5)

glVertex3f(0.5, -0.5, 0.5)

glVertex3f(-0.5, -0.5, 0.5)

glVertex3f(-0.5, 0.5, 0.5)

glEnd()

glFlush()

-

----- INICIA O OPENGL EM MODO POLIGONO

----- DEFINE A COR DE ESCRITA VERMELHO

----- CRIA UM VERTICE NO PONTO DET.

----- CRIA UM VERTICE NO PONTO DET.

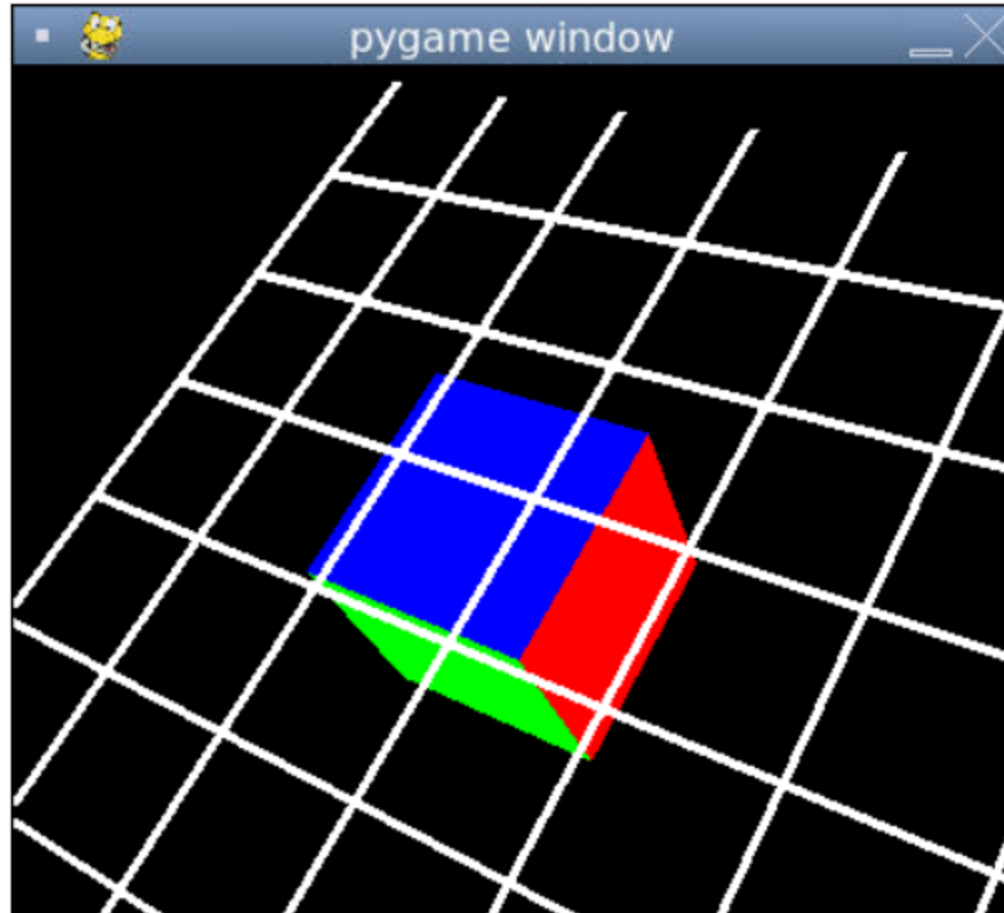
----- CRIA UM VERTICE NO PONTO DET.

----- CRIA UM VERTICE NO PONTO DET.

----- FINALIZA A ESCRITA

----- LIMPA O BUFFER

Programa - Código (MAIN)



Exercício

Utilize como base os exemplos anteriores.

Construa o modelo e rode a aplicação.

Crie um triangulo e rotacione o mesmo.

ATENÇÃO

Não basta copiar!!!

Entender é mais importante!!!!



Instruções

Utilize como base os exemplos anteriores.

ATENÇÃO

Não basta copiar!!!

Entender é mais importante!!!!



Regras da Entrega

Arquivo em PDF contendo:

- Nome e Número de matrícula
- Código fonte
- Tela capturada com o resultado