

UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD
DEL CUSCO

FACULTAD DE INGENIERÍA ELÉCTRICA,
ELECTRÓNICA, INFORMÁTICA Y MECÁNICA
INGENIERÍA INFORMÁTICA Y DE SISTEMAS



Guía de Laboratorio 7 - Deformación 2D

Alumno:

Ian Logan Will Quispe Ventura

211359

Docente:

Hector Eduardo Ugarte Rojas

Curso:

Computación Gráfica

Cusco - Perú
2023 - II

Deformación de un rectángulo en el eje X

```
from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *

# Deformación de rectángulo con respecto al eje X
def deformaRectanguloX(x1, y1, x2, y2, sh):

    #--dibujar el rectangulo original
    glColor3f (0.5 , 0.3 , 0.9) # Color1
    glBegin(GL_LINE_LOOP)
    glVertex2f(x1, y1)
    glVertex2f(x2, y1)
    glVertex2f(x2, y2)
    glVertex2f(x1, y2)
    glEnd()
    # Deformar cada punto del rectangulo con respecto
    al eje X
    x1d = x1 + sh * y1
    y1d = y1
    x2d = x2 + sh * y1
    y2d = y1

    x3d = x2 + sh * y2
    y3d = y2
    x4d = x1 + sh * y2
    y4d = y2
    # Dibujar el rectangulo deformado con respecto al
    eje X
    glColor3f(0.8, 0.26, 1.0) # Color2
    glBegin(GL_LINE_LOOP)
    glVertex2f(x1d, y1d)
    glVertex2f(x2d, y2d)
    glVertex2f(x3d, y3d)
    glVertex2f(x4d, y4d)
    glEnd()
```

```

#--despliega el gráfico
def display():
    x1 = 20.0
    y1 = 50.0
    x2 = 250.0
    y2 = 200.0
    sh = 2
    glClear(GL_COLOR_BUFFER_BIT)
    deformaRectanguloX(x1, y1, x2, y2, sh)
    glFlush()

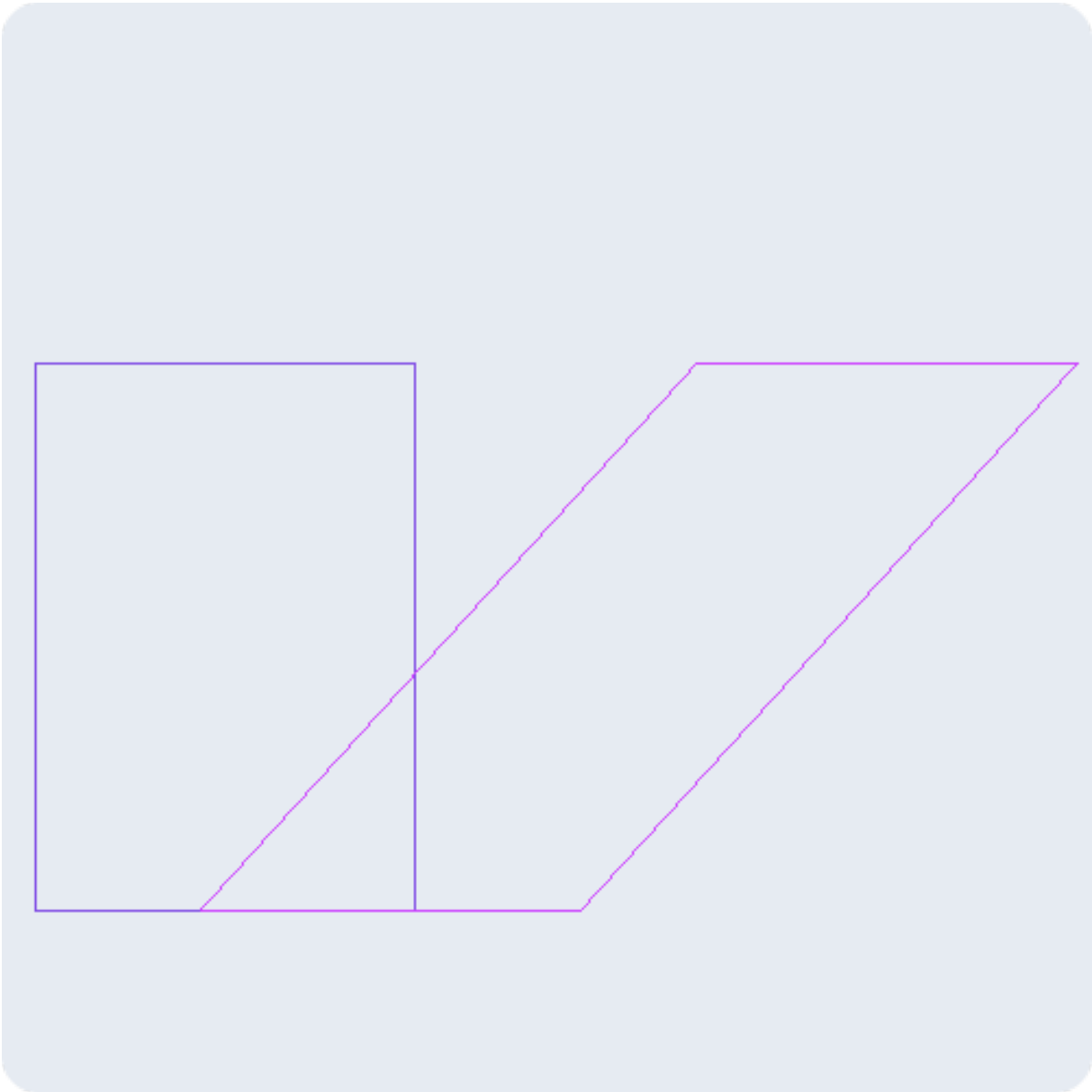
def myinit():
    glClearColor (0.9 ,0.92 , 0.95 , 1.0) # Fondo
    glPointSize(1.0) #--tamaño de los puntos
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluOrtho2D(0.0, 659.0, 0.0, 299.0)

def main():
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB)
    glutInitWindowSize(500, 500)
    glutInitWindowPosition(0, 0)
    glutCreateWindow("Reflexión")
    glutDisplayFunc(display)
    myinit() # fija
    glutMainLoop()

if __name__ == "__main__":
    main()

```

Gráfico generado



Deformación de un rectángulo en el eje Y

```
from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *
import sys

#--deformación de rectángulo con respecto al eje X
def deformaRectanguloX(x1, y1, x2, y2, sh):
    #Dibujar el rectangulo original
    glColor3f (0.5 , 0.3 , 0.9) # Color1
    glBegin(GL_LINE_LOOP)
    glVertex2f(x1, y1)
    glVertex2f(x2, y1)
    glVertex2f(x2, y2)
    glVertex2f(x1, y2)
    glEnd()
    # Deformar cada punto del rectangulo con respecto
    # al eje X
    x1d = x1
    y1d = y1 + sh * x1
    x2d = x2
    y2d = y1 + sh * x2

    x3d = x2
    y3d = y2 + sh * x2
    x4d = x1
    y4d = y2 + sh * x1
    # Dibujar el rectangulo deformado con respecto al
    # eje X
    glColor3f(0.8, 0.26, 1.0) # Color2
    glBegin(GL_LINE_LOOP)
    glVertex2f(x1d, y1d)
    glVertex2f(x2d, y2d)
    glVertex2f(x3d, y3d)
    glVertex2f(x4d, y4d)
    glEnd()
```

```

#--despliega el gráfico
def display():
    x1 = 20.0
    y1 = 50.0
    x2 = 230.0
    y2 = 200.0
    sh = 2
    glClear(GL_COLOR_BUFFER_BIT)
    deformaRectanguloX(x1, y1, x2, y2, sh)
    glFlush()

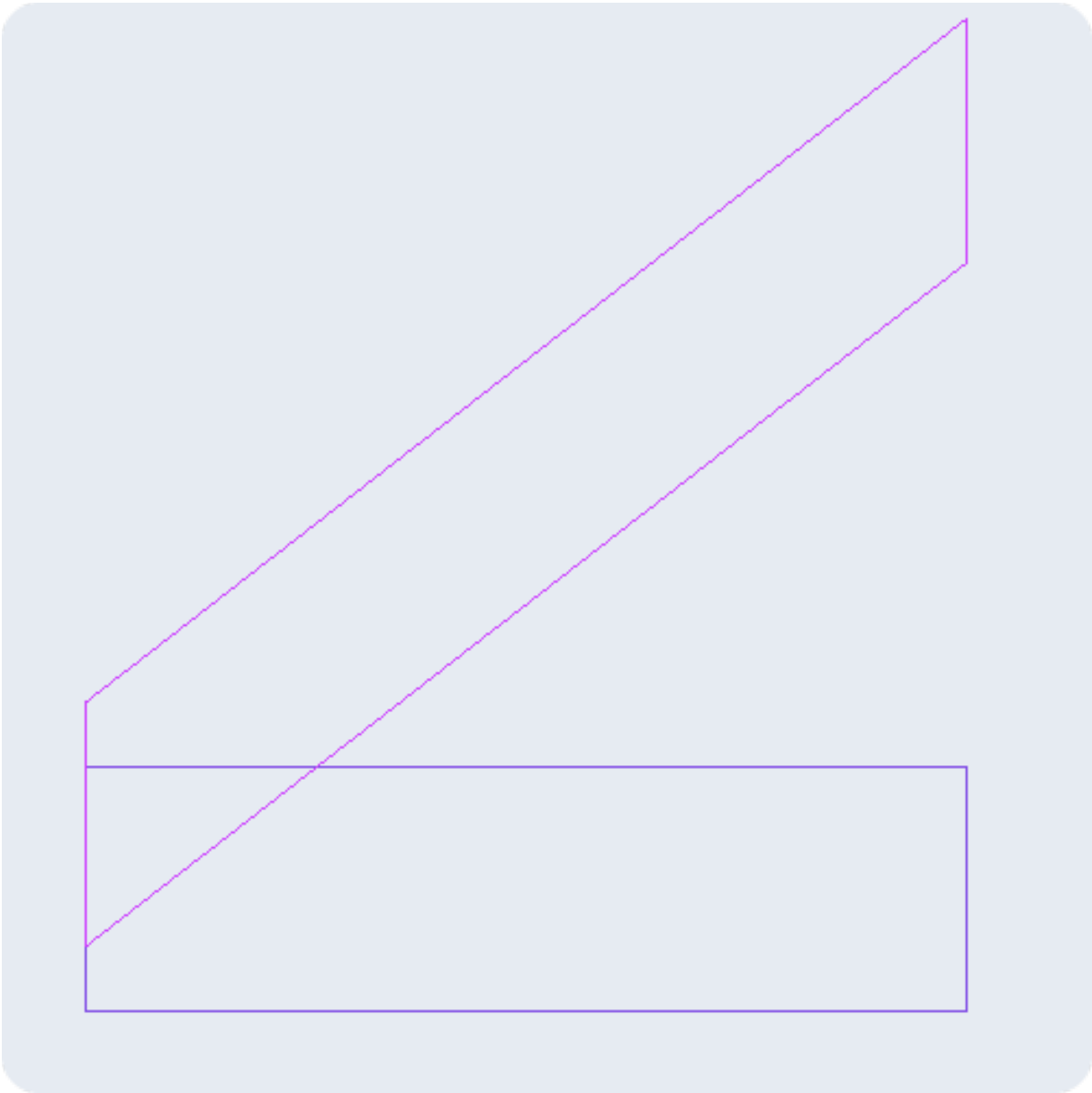
def myinit():
    glClearColor (0.9 ,0.92 , 0.95 , 1.0) #
        Fondo
    glPointSize(1.0)
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluOrtho2D(0.0, 260.0, 0.0, 670.0)

def main():
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_SINGLE |
        GLUT_RGB)
    glutInitWindowSize(500, 500)
    glutInitWindowPosition(0, 0)
    glutCreateWindow("Reflexión")
    glutDisplayFunc(display)
    myinit()
    glutMainLoop()

if __name__ == "__main__":
    main()

```

Gráfico generado



Deformación de "E" en el eje X

```
from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *
import sys

# Matriz de coordenadas del dibujo original
matriz_origen = [[200, 300, 300, 300, 300, 280, 200,
                  280],
                 [200, 290, 230, 290, 230, 180, 200, 180],
                 [200, 250, 260, 250, 260, 230, 200, 230],
                 [200, 200, 300, 200, 300, 180, 200, 180]]

# Dibujar la figura original
def dibujar_E():
    glBegin(GL_QUADS)
    glColor3f(0.5, 0.3, 0.9) # Color1
    for i in range(4):
        for j in range(0, 8, 2):
            glVertex2f(matriz_origen[i][j],
                      matriz_origen[i][j+1])
    glEnd()

# Algoritmo de deformación
def deformaPuntoX(x, y, sh):
    return x + sh * y, y

# Dibujar la figura E deformada respecto a x
def dibujar_E_deformado(sh):
    glBegin(GL_QUADS)
    glColor3f(0.8, 0.26, 1.0) # Color2
    for i in range(4):
        for j in range(0, 8, 2):
            x, y = deformaPuntoX(matriz_origen[i][j],
                                  matriz_origen[i][j+1], sh)
            glVertex2f(x, y)
    glEnd()
```



```

def display():
    glClearColor(GL_COLOR_BUFFER_BIT)
    dibujar_E()
    dibujar_E_deformado(2)
    glFlush()
def display():
    glClearColor(GL_COLOR_BUFFER_BIT)
    dibujar_E()
    dibujar_E_deformado(2)
    glFlush()

def myinit():
    glClearColor (0.9 ,0.92 , 0.95 , 1.0)
    glPointSize(1.0)
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluOrtho2D(0.0, 999.0, 0.0, 499.0)

def main():
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB)
    glutInitWindowSize(500, 500)
    glutInitWindowPosition(0, 0)
    glutCreateWindow("Reflexión")
    glutDisplayFunc(display)
    myinit()
    glutMainLoop()

if __name__ == "__main__":
    main()

```

Gráfico generado



Deformación de "E" en el eje Y

```
from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *
import sys

# Matriz de coordenadas del dibujo original
matriz_origen = [[200, 300, 300, 300, 300, 280, 200,
                  280],
                 [200, 290, 230, 290, 230, 180, 200, 180],
                 [200, 250, 260, 250, 260, 230, 200, 230],
                 [200, 200, 300, 200, 300, 180, 200, 180]]

# Dibujar la figura original
def dibujar_E():
    glBegin(GL_QUADS)
    glColor3f(0.5, 0.3, 0.9) # Color1
    for i in range(4):
        for j in range(0, 8, 2):
            glVertex2f(matriz_origen[i][j],
                      matriz_origen[i][j+1])
    glEnd()

# Algoritmo de deformación
def deformaPuntoY(x, y, sh):
    return y + sh * x, x

# Dibujar la figura E deformada respecto a y
def dibujar_E_deformado(sh):
    glBegin(GL_QUADS)
    glColor3f(0.8, 0.26, 1.0) # Color2
    for i in range(4):
        for j in range(0, 8, 2):
            x, y = deformaPuntoY(matriz_origen[i][j],
                                  matriz_origen[i][j+1], sh)
            glVertex2f(x, y)
    glEnd()
```

```

def display():
    glClear(GL_COLOR_BUFFER_BIT)
    dibujar_E()
    dibujar_E_deformado(2)
    glFlush()
def myinit():
    glClearColor (0.9 ,0.92 , 0.95 , 1.0) #
        Fondo
    glPointSize(1.0)
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluOrtho2D(0.0, 999.0, 0.0, 499.0)

def main():
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_SINGLE |
        GLUT_RGB)
    glutInitWindowSize(500, 500)
    glutInitWindowPosition(0, 0)
    glutCreateWindow("Reflexión")
    glutDisplayFunc(display)
    myinit()
    glutMainLoop()

if __name__ == "__main__":
    main()

```

Gráfico generado

