UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO

FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA,INFORMÁTICA Y MECÁNICA

INGENIERÍA INFORMÁTICA Y DE SISTEMAS

Guía de Laboratorio 8 - Composición de Transformaciones

Alumno:
Ian Logan Will Quispe Ventura
211359

Docente:
Hector Eduardo Ugarte Rojas

Curso:
Computación Gráfica

Cusco - Perú
2023 - II

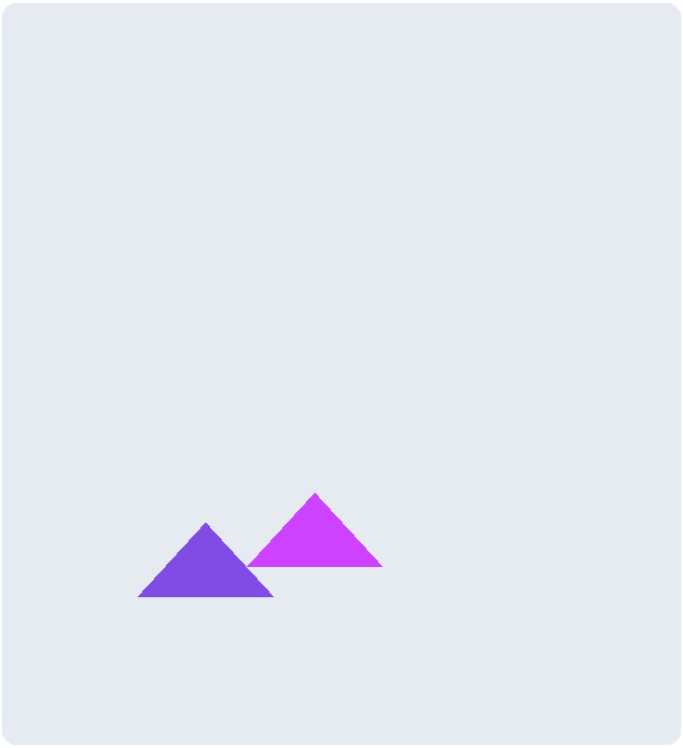
Traslación, Rotación, Escalado y Transformación de un triángulo

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

def Traslada():
    # Triangulo antes de trasladar
    glBegin(GL_TRIANGLES)
    glColor3f (0.5 , 0.3 , 0.9) # Color1
    glVertex2f(100.0, 100.0)
    glVertex2f(200.0, 100.0)
    glVertex2f(150.0, 150.0)
    glEnd()
    # Triangulo después de trasladar
    glTranslatef(200.0, 200.0, 0.0)
    glBegin(GL_TRIANGLES)
    glColor3f(0.8, 0.26, 1.0) # Color2
    glVertex3f(100.0, 100.0, 0.0)
    glVertex3f(200.0, 100.0, 0.0)
    glVertex3f(150.0, 150.0, 0.0)
    glEnd()

def Rota():
    #--triangulo antes de rotar
    glColor3f(0.5, 1.0, 0.7)
    glBegin(GL_TRIANGLES)
    glVertex2f(100.0, 100.0)
    glVertex2f(200.0, 100.0)
    glVertex2f(150.0, 150.0)
    glEnd()
    #--triangulo rotado
    glRotatef(45, 0, 0, 1)
    glBegin(GL_TRIANGLES)
    glVertex2f(100.0, 100.0)
    glVertex2f(200.0, 100.0)
    glVertex2f(150.0, 150.0)
    glEnd()
```

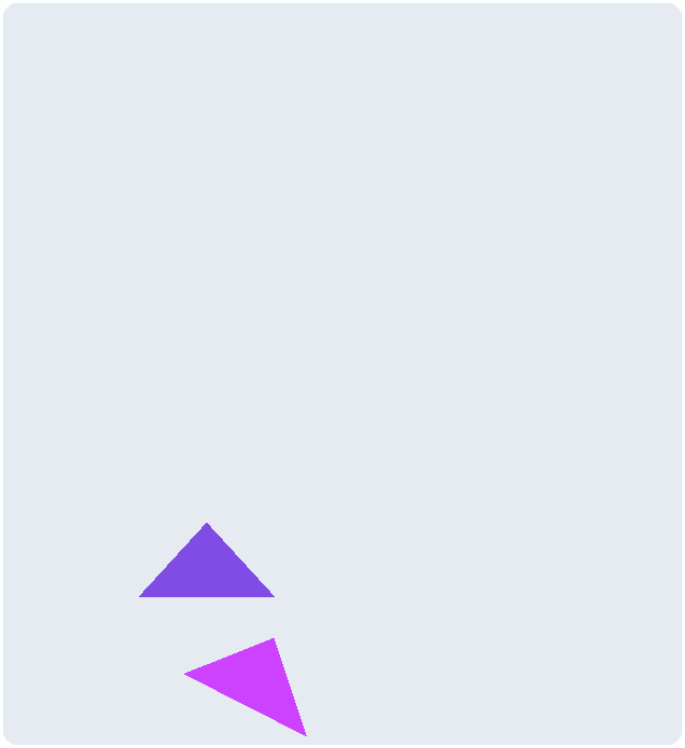
```
def Escala():
    #--triangulo antes de escalar
    glColor3f(0.5, 1.0, 0.7)
    glBegin(GL_TRIANGLES)
    glVertex2f(100.0, 100.0)
    glVertex2f(200.0, 100.0)
    glVertex2f(150.0, 150.0)
    glEnd()
    #--triangulo despues de escalar
    glScalef(2.0, 2.0, 2.0)
    glBegin(GL_TRIANGLES)
    glVertex2f(100.0, 100.0)
    glVertex2f(200.0, 100.0)
    glVertex2f(150.0, 150.0)
    glEnd()
def TransfComp():
    #--triangulo antes de rotar y trasladar
    glColor3f(0.5, 1.0, 0.7)
    glBegin(GL_TRIANGLES)
    glVertex2f(100.0, 100.0)
    glVertex2f(200.0, 100.0)
    glVertex2f(150.0, 150.0)
    glEnd()
    #--triangulo después de rotar y trasladar
    glRotatef(45, 0, 0, 1)
    glTranslatef(200.0, 200.0, 0.0)
    glBegin(GL_TRIANGLES)
    glVertex2f(100.0, 100.0)
    glVertex2f(200.0, 100.0)
    glVertex2f(150.0, 150.0)
    glEnd()
    glFlush()
def display():
    glClear(GL_COLOR_BUFFER_BIT)
    Traslada()
    Rota()
    Escala()
    TransfComp()
    glFlush()
```

Gráficos generados

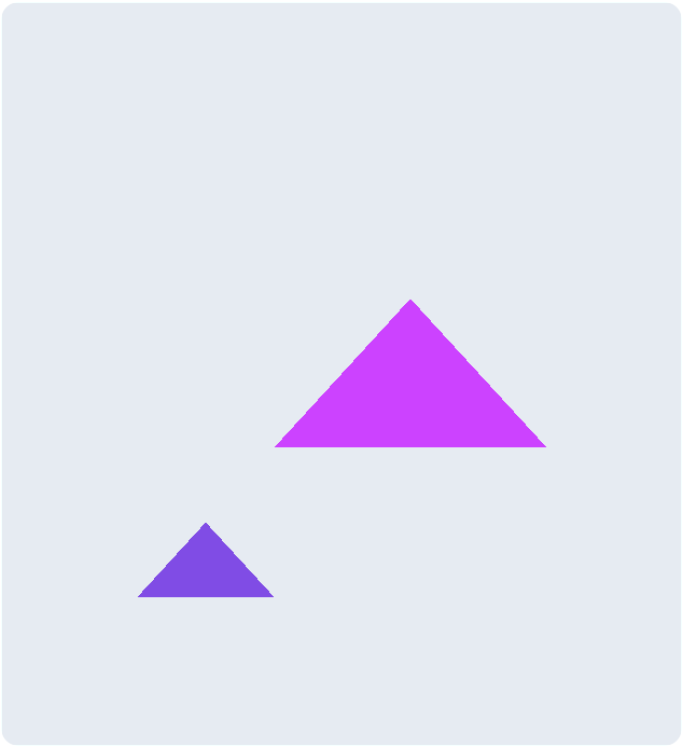
Traslación



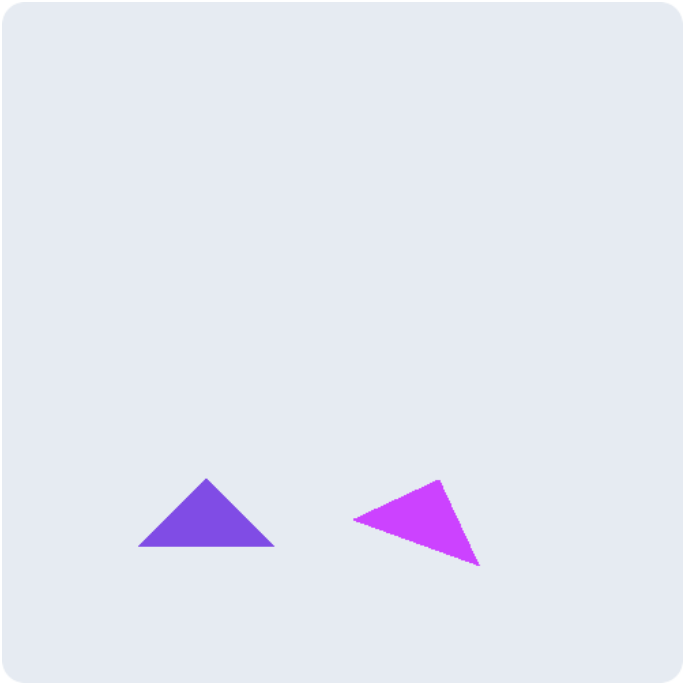
Rotación



Escalado



Transformacion



Animación de la rotación y traslación de un cuadrado

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

angle = 0.0
def initGL():
    glClearColor (0.9 ,0.92 , 0.95 , 1.0) #
      Fondo

def idle():
    global angle
    angle += 0.2
    glutPostRedisplay()

def display():
    global angle
    glClear(GL_COLOR_BUFFER_BIT)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()
    glPushMatrix()

    # Rotar y trasladar
    glTranslatef(-0.5, 0.4, 0.0)
    glRotatef(angle, 0.0, 0.0, 1.0)

    glBegin(GL_QUADS)
    glColor3f (0.5 , 0.3 , 0.9) # Color1
    glVertex2f(-0.3, -0.3)
    glVertex2f(0.3, -0.3)
    glVertex2f(0.3, 0.3)
    glVertex2f(-0.3, 0.3)
    glEnd()
    glPopMatrix()
    glPushMatrix()
    # Rotar y trasladar
    glTranslatef(-0.4, -0.3, 0.0)
    glRotatef(angle, 0.0, 0.0, 1.0)
```

```
    glBegin(GL_QUADS)
    glColor3f(0.8, 0.26, 1.0) # Color2
    glVertex2f(-0.3, -0.3)
    glVertex2f(0.3, -0.3)
    glVertex2f(0.3, 0.3)
    glVertex2f(-0.3, 0.3)
    glEnd()
    glPopMatrix()
    glutSwapBuffers()

def main():
    import sys
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_DOUBLE)
    glutInitWindowSize(540, 480)
    glutInitWindowPosition(50, 50)
    glutCreateWindow("Animation via Idle Function".
      encode("ascii"))
    glutDisplayFunc(display)
    glutIdleFunc(idle)
    initGL()
    glutMainLoop()

if __name__ == "__main__":
    main()
```

Gráfico generado

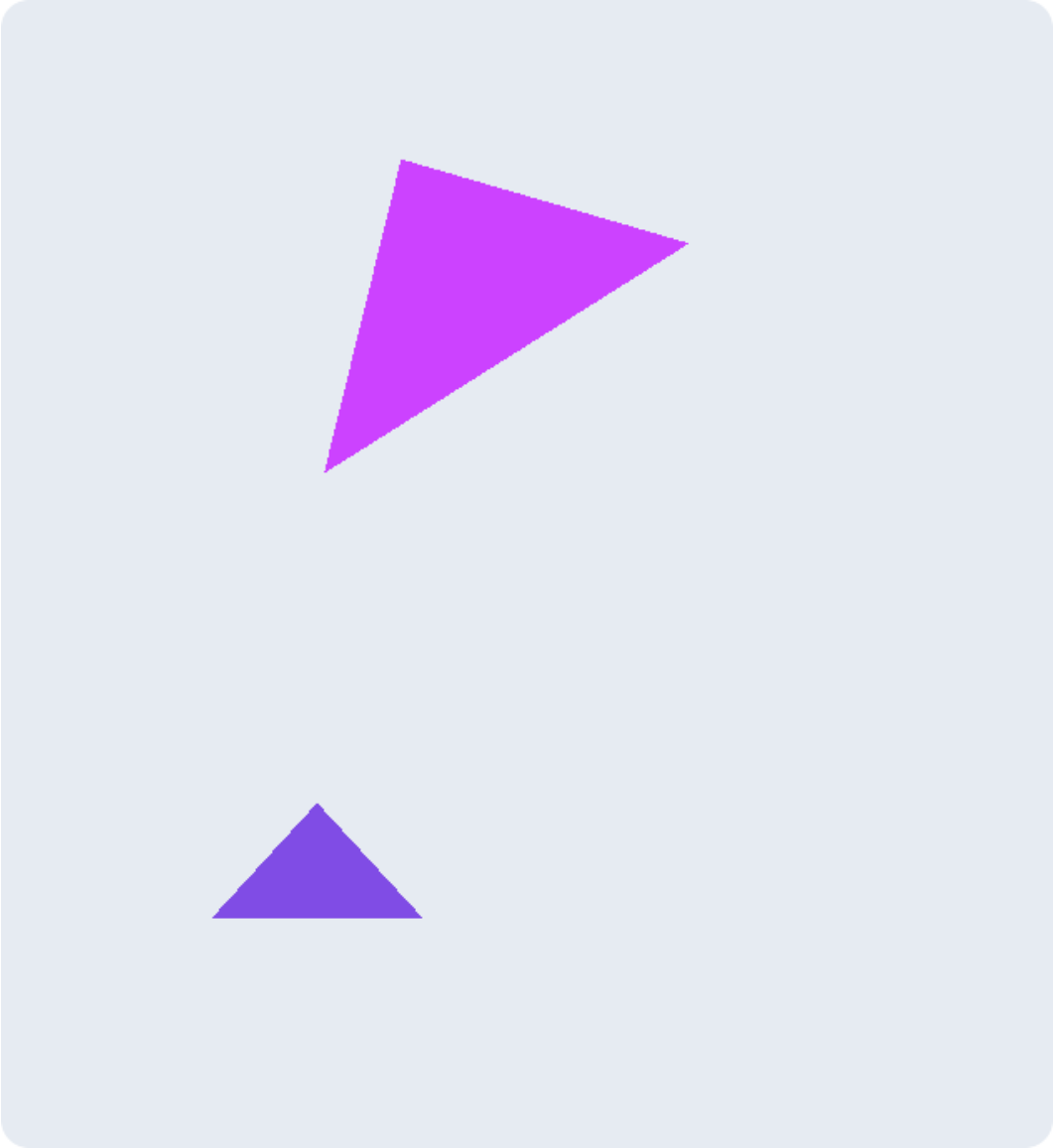
Animación GIF: `https://gifyu.com/image/S06e4`

Escadalo (x2), Rotación (30°) y Traslacion (80, 20) de un triángulo

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

def Transformar():
    # Triangulo antes de rotar y trasladar
    glColor3f (0.5 , 0.3 , 0.9) # Color1
    glBegin(GL_TRIANGLES)
    glVertex2f(100.0, 100.0)
    glVertex2f(200.0, 100.0)
    glVertex2f(150.0, 150.0)
    glEnd()

    # Triangulo después de escalar, rotar y trasladar
    glTranslatef(80.0, 20.0, 0.0)
    glRotatef(30, 0, 0, 1)
    glScalef(2.0, 2.0, 0)

    glBegin(GL_TRIANGLES)
    glColor3f(0.8, 0.26, 1.0) # Color2
    glVertex2f(100.0, 100.0)
    glVertex2f(200.0, 100.0)
    glVertex2f(150.0, 150.0)
    glEnd()
    glFlush()

def display():
    glClear(GL_COLOR_BUFFER_BIT)
    Transformar()
    glFlush()

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

```
def main():
    glutInit()
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB)
    glutInitWindowSize(500, 500)
    glutInitWindowPosition(0, 0)
    glutCreateWindow("Transformaciones con OpenGL")
    glutDisplayFunc(display)
    myinit()
    glutMainLoop()

if __name__ == "__main__":
    main()
```

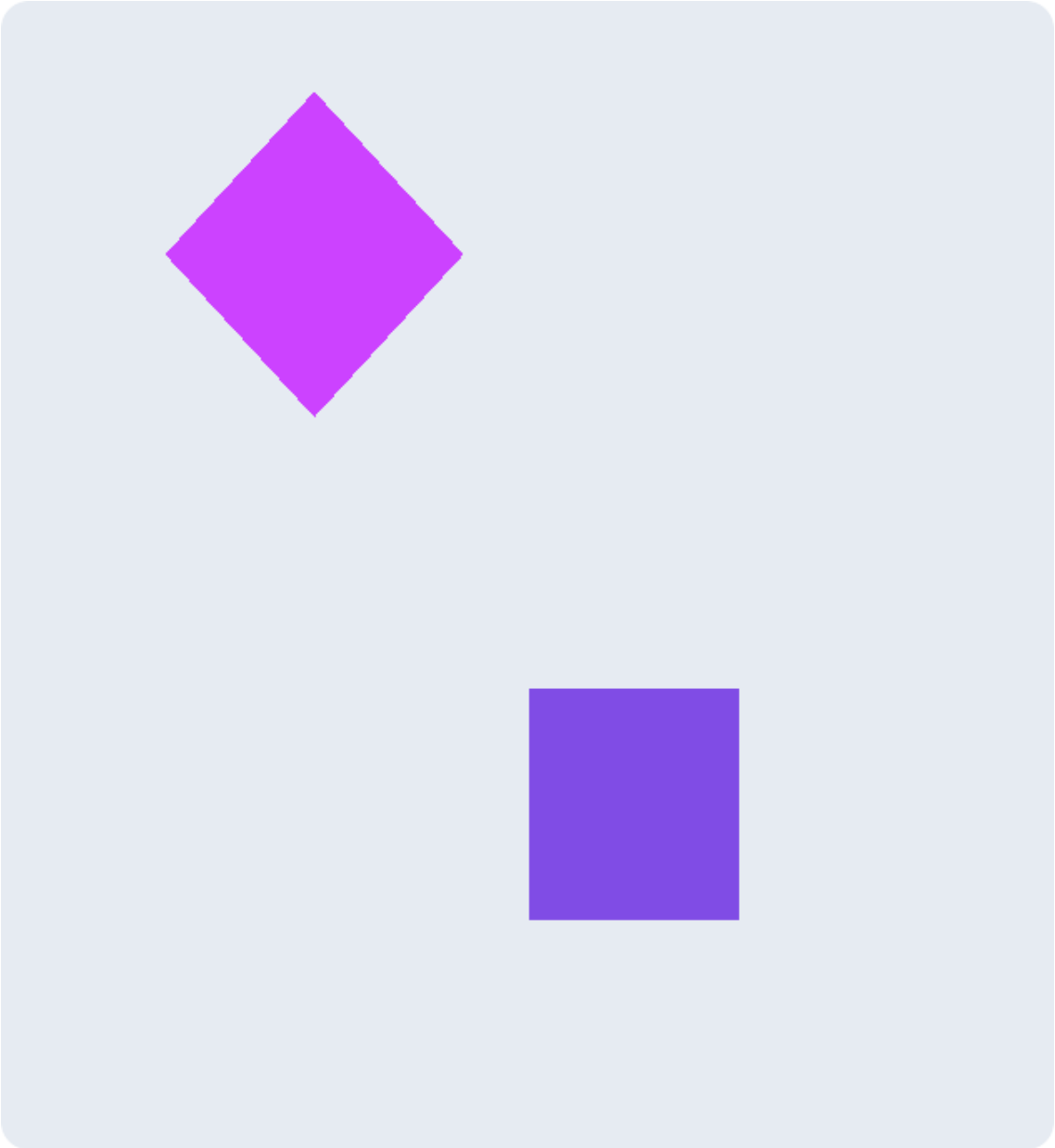
Gráfico generado



Traslación y Rotación de un cuadrado

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

def cuadrado():
    # Dibujar el Cuadrado
    glBegin(GL_QUADS)
    glVertex2f(250.0, 100.0)
    glVertex2f(350.0, 100.0)
    glVertex2f(350.0, 200.0)
    glVertex2f(250.0, 200.0)
    glEnd()

def display():
    glClear(GL_COLOR_BUFFER_BIT)
    glColor3f (0.5 , 0.3 , 0.9) # Color1
    cuadrado()
    # Trasladar y rotar el cuadrado
    glRotatef(45, 0, 0, 1)
    glTranslatef(80.0, 20.0, 0.0)
    glColor3f(0.8, 0.26, 1.0) # Color2
    cuadrado()
    glFlush()
def myinit():
    glClearColor (0.9 ,0.92 , 0.95 , 1.0) # Fondo
    glPointSize(1.0)
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluOrtho2D(0.0, 499.0, 0.0, 499.0)
def main():
    glutInit()
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB)
    glutInitWindowPosition(0, 0)
    glutCreateWindow("Transformaciones con OpenGL")
    glutDisplayFunc(display)
    myinit()
    glutMainLoop()
```

Gráfico generado



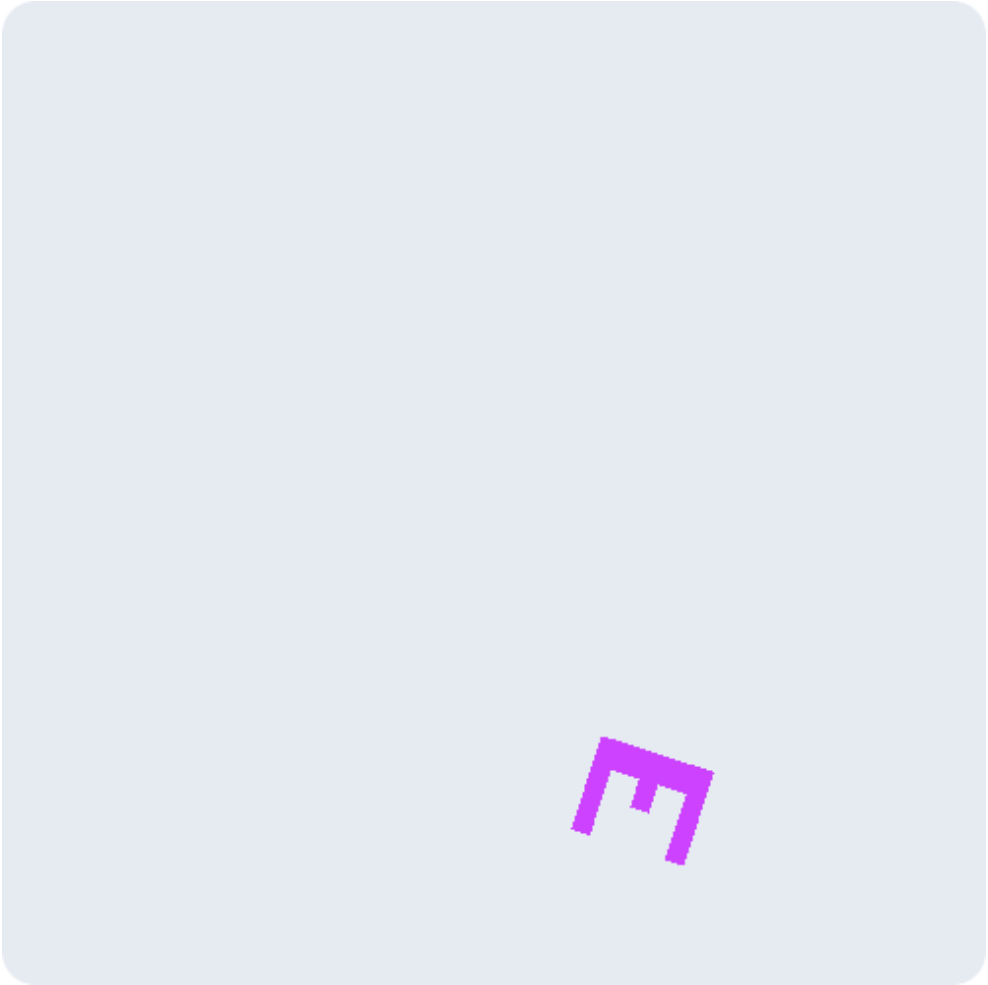
Animación de la rotación de la figura "E"

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

# Global variable
angle = 0.0  # Current rotational angle of the shapes

# 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]]

def dibujar_E():
    glBegin(GL_QUADS)
    glColor3f(0.8, 0.26, 1.0)
    glVertex2f(matriz_origen[0][0],matriz_origen
      [0][1])
    glVertex2f(matriz_origen[0][2],matriz_origen
      [0][3])
    glVertex2f(matriz_origen[0][4],matriz_origen
      [0][5])
    glVertex2f(matriz_origen[0][6],matriz_origen
      [0][7])

    glVertex2f(matriz_origen[1][0],matriz_origen
      [1][1])
    glVertex2f(matriz_origen[1][2],matriz_origen
      [1][3])
    glVertex2f(matriz_origen[1][4],matriz_origen
      [1][5])
    glVertex2f(matriz_origen[1][6],matriz_origen
      [1][7])
```

```
    glVertex2f(matriz_origen[2][0],matriz_origen
       [2][1])
    glVertex2f(matriz_origen[2][2],matriz_origen
       [2][3])
    glVertex2f(matriz_origen[2][4],matriz_origen
       [2][5])
    glVertex2f(matriz_origen[2][6],matriz_origen
       [2][7])

    glVertex2f(matriz_origen[3][0],matriz_origen
       [3][1])
    glVertex2f(matriz_origen[3][2],matriz_origen
       [3][3])
    glVertex2f(matriz_origen[3][4],matriz_origen
       [3][5])
    glVertex2f(matriz_origen[3][6],matriz_origen
       [3][7])
    glEnd()

def initGL():
    glClearColor (0.9 ,0.92 , 0.95 , 1.0) # Fondo
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluOrtho2D(-499.0, 499.0, -499.0, 499.0)
def idle():
    global angle
    angle += 0.9
    glutPostRedisplay()
def display():
    global angle
    glClear(GL_COLOR_BUFFER_BIT)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()
    glPushMatrix()
    glTranslatef(-0.25, -0.24, 0.0)
    glRotatef(angle, 0.0, 0.0, 1.0)
    dibujar_E()
    glPopMatrix()
    glutSwapBuffers()
```

```
def main():
    import sys
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_DOUBLE)
    glutInitWindowSize(500, 500)
    glutInitWindowPosition(50, 50)
    glutCreateWindow("Animation via Idle Function".
      encode("ascii"))
    glutDisplayFunc(display)
    glutIdleFunc(idle)
    initGL()
    glutMainLoop()

if __name__ == "__main__":
    main()
```

Gráfico generado



Animación GIF: `https://gifyu.com/image/S06eY`