

# CURSO: CC322 - 2017 -2

## Practica 2. Sistema de ventanas GLUT, Primitivas Básicas.

### 1. Introducción

#### Objetivo general:

- Conocer y utilizar el sistema de ventanas GLUT y algunas primitivas básicas.
  - Al final de la practica, generará un archivo con los programas que haya modificado
- El nombre del archivo será <Nombre-Apellido-lab02.zip>

#### Objetivos particulares:

- Desarrollar ejemplos con el uso de GLUT
- Desarrollar ejemplos con Primitivas Básicas

### 2 . Recursos Informáticos

<https://www.opengl.org>

[https://www.khronos.org/opengl/wiki/Getting\\_Started](https://www.khronos.org/opengl/wiki/Getting_Started)

[https://www.khronos.org/opengl/wiki/Category:Core\\_API\\_Reference](https://www.khronos.org/opengl/wiki/Category:Core_API_Reference)

<https://www.python.org/>

<https://www.jetbrains.com/pycharm/>

### 3. DESARROLLO

1. Verifique si tiene instalado Python. Si no está Proceda a instalarlo

```
sudo apt-get install python2.7  
type python3 python2.7 python3.5 python2 python  
sudo apt-get install python-opengl
```

2. Verifique si tiene instalado pip. Si no está proceda a instalarlo

```
sudo apt-get install python-pip
```

3. Verifique si tiene un editor de texto adecuado para editar programas en Python

(p.ej. Geany, SublimeText, u otro con el que esté familiarizado) si no está – Proceda a Instalarlo.

Se recomienda PyCharm , version community - <https://www.jetbrains.com/pycharm/>

- 3.Verifique si tiene instaladas las librerías OPENGGL Y GLUT. Si no están proceda a instalarlas.

```
sudo apt-get install freeglut3-dev
```

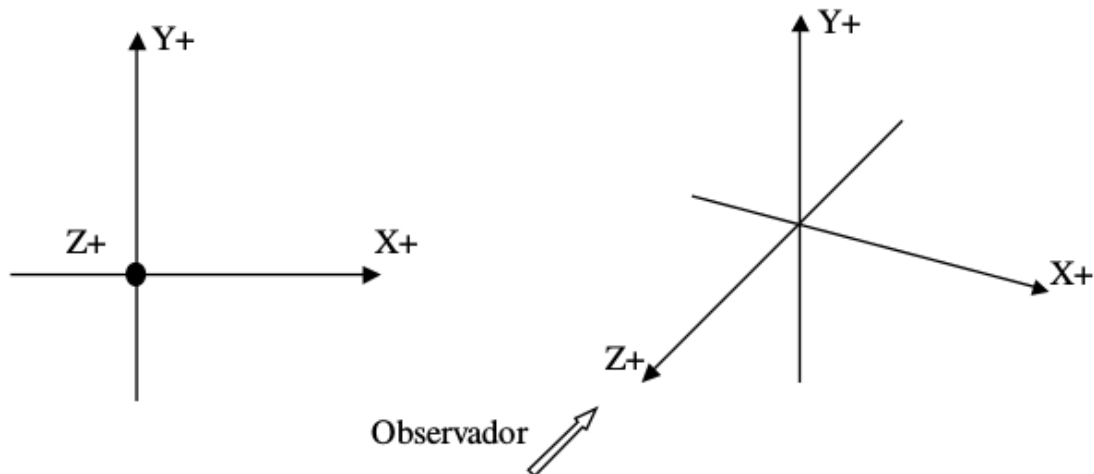
## 4 . Ejemplos Prácticos

### Ejemplo 4.1.

Copie el siguiente fragmento en un archivo con extension “.py”, p.ej. *lab02-1.py*

Trate de correrlo y verifique que no haya errores de tipeo ni de otra indole.

Recuerde que en OpenGL el modelo del sistema de coordenadas es el siguiente:



```
# lab02-1.py
# Dibujando con OpenGL Parte 1
from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *
import sys

# La siguiente es una funcion de inicializacion
def init():
    # utilizaremos el color negro como fondo
    # los numeros (de 0 a 1.0) corresponden a (Red,Green,Blue,alpha)
    glClearColor(0.0, 0.0, 0.0, 1.0)
    # Definimos los rangos del sistema de coordenadas
    # gluOrtho2D(x-izquierda, x-derecha, y-abajo, y-arriba)
    # lo que coloca el origen (0,0) en el centro de la pantalla
    gluOrtho2D(-1.0, 1.0, -1.0, 1.0)

# La siguiente funcion presenta la escena
def display():
    # la siguiente funcion limpia la pantalla y pinta el fondo
    glClearColor(1.0, 1.0, 1.0, 0.0);
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glColor3f(1.0, 0.0, 0.0);
    glutWireTorus(0.25, 0.75, 28, 28);
    glColor3f(0.0, 0.0, 1.0);
    glutWireCube(.60);
    glutSwapBuffers();
    # Levantamos el lapiz - dejamos de dibujar
    # glEnd()
    # presentamos la escena en pantalla
    glFlush()

#La funcion reshape se llama cada vez que se redimensiona la ventana
```

```

#Los parametros que recibe son el ancho y la altura de la ventana
def reshape(width, height):
    # glViewport Define la porcion de ventana donde OpenGL podra dibujar
    #Parametros: Dist Horiz, Dist vert del inicio (Esq.Sup.Izquierda),
    #ancho y altura de la ventana
    glViewport(0, 0, width, height);
    #glMatrixMode Define la matriz de transformacion elegida
    #Existen:GL_PROJECTION - proyeccion ,GL_MODELVIEW - Modelado,
    # GL_TEXTURE -Texturas
    glMatrixMode(GL_PROJECTION);
    #glLoadIdentity carga como matriz de proyeccion la matriz identidad - la
inicializa
    glLoadIdentity();
    #gluPerspective opera sobre la matriz de proyeccion y
    #define el angulo del campo de vision en sentido vertical (en grados)
    #la relacion entre la altura y la anchura de la figura (aspecto)
    #los planos: el mas cercano y el mas distante a la camara
    gluPerspective(60.0, height / width, 1.0, 128.0);
    # Para las siguientes operaciones se elige la matriz de Modelado
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity()
    #gluLookAt define la transformacion sobre la vista inicial.
    #tiene 9 parametros: los primeros tres representan la
    #distancia en x, y, z de los ojos del observador
    #los siguientes tres, las coordenadas x,y, z del punto de referencia a
observar
    # y los ultimos tres, la direccion del upVector
    gluLookAt(0.0, 1.0, 2.0, 0.0, 0.5, 0.5, 0.0, 1.0, 0.0);
# usaremos la funcion main para iniciar OPENGL y llamar
# rutinas de inicializacion como init
def main():
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB)
    glutInitWindowSize(500, 500)
    glutInitWindowPosition(50, 50)
    glutCreateWindow("Laboratorio 2")
    glutDisplayFunc(display)
    init()
    glutReshapeFunc(reshape)
    glutMainLoop()
main()
# End of Program

```

4.1 Dibuje un ovni con las figuras de unToroide y una Esfera.

4.2 Modificar la función glViewport de manera que al alargar la ventana la figura no se deforme. Se logra haciendo que el viewport sea siempre cuadrada, de dimensión el menor de los valores de la altura y la anchura. El valor de la relación entre la altura y la anchura para la función gluPerspective() es ahora siempre 1.

4.3 Probar diferentes vistas iniciales con la función gluLookAt.

4.4 Modifique el ejemplo anterior (en un archivo diferente) para que se pueda utilizar el teclado para rotar la figura y salir del modo de ejecución del programa.

Para esto, agregue la funcion **glutKeyboardFunc(keyboard)** en la funcion main(). La función keyboard() que es pasada como parámetro será llamada cada vez que ocurra un evento en el teclado. Se define a continuación la función keyboard():

```
def keyboard(key, x, y):
    print key
    if (key == 'h'):
        print("help")
        print("c - switch culling")
        print("q/escape - Salir")
        print("1/2 - rotar")
    elif (key == 'c'):
        if (glIsEnabled(GL_CULL_FACE)):
            glDisable(GL_CULL_FACE)
            # print 'was enabled'
        else:
            glEnable(GL_CULL_FACE)
            # print 'was disabled'
    elif (key == '1'):
        glRotatef(1.0, 1., 0., 0.)
    elif (key == '2'):
        glRotatef(1.0, 0., 1., 0.)
    elif (key == 'q' or key == chr(27)):
        exit(0)
    # Esta funcion indica a la GLUT que es necesario redibujar la ventana
    glutPostRedisplay()
```

Los parámetros de **glRotatef** corresponden al ángulo a rotar y a los componentes x, y, z del eje sobre el que va a rotar la figura

4.5. Modifique el ejemplo anterior (en un archivo diferente) y dibuje una tortuga con esferas utilizando un código parecido al siguiente:

```
glPushMatrix()
glTranslatef(0.25, 0.0, 0.0)
glutWireSphere(0.6, 15, 15)
glTranslatef(0.25, 0.0, 0.0)
glutWireSphere(0.6, 15, 15)
glPopMatrix()
```

4.6 Modifique el programa anterior en un archivo diferente para utilizar el mouse

- inicialice una variable spin en 0  
el valor de spin cambiara con la accion del boton del mouse.
- haga una transformacion para rotar la escena con  
glRotatef(1.0, spin, 0., 0.) en la funcion que dibuja la escena.
- En la Funcion principal, llame a la funcion glutMouseFunc(mousefunction) para monitorear eventos del mouse
- defina la funcion mousefunction asi:

```
def mousefunction(button, state, x, y):
    global spin
    if button == GLUT_LEFT_BUTTON and state == GLUT_DOWN:
        spin = (spin + 30.0) % 360
        glutPostRedisplay()
```