

CURSO: CC322 - 2017-2

Practica 1. Inicializacion de un entorno de desarrollo y primeros ejemplos.

1. Introducción

Objetivo general:

-Conocer y utilizar las algunas de las principales herramientas y librerias a utilizarse en el curso.

Objetivos particulares:

- Configurar el entorno de desarrollo
- Verificar si estan correctamente instaladas las librerias requeridas para la practica con OPENGL
- Desarrollar ejemplos para comprobar la validez del entorno de desarrollo

2 . Recursos Informáticos

<https://www.opengl.org>

https://www.khronos.org/opengl/wiki/Getting_Started

https://www.khronos.org/opengl/wiki/Category:Core_API_Reference

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

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
```

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.

3.Verifique si tiene instaladas las librerias OPENGL 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. Ogl1.py
Trate de correrlo y verifique que no haya errores de tipeo ni de otra indole.

```
#ejemplo1.py

# Super Numeros

import string

i = input("Por favor ingrese el limite superior")

for n in range(i):

    x= 3*n**3

    if string.find(str(x), "333")<> -1:

        print n, " ", x

#fin de programa
```

Cuando el programa lo solicite ingrese un número (p.Ej. 1000)

4.1 Cual es el resultado?

4.2 Que Hace el programa?

4.3 Explique lo que hace cada linea de código.

4.4 Que valores toma la funcion string.find ?

4.5 Si llamamos a este programa Super-3 , Halle el minimo n para el programa Super-5 y presente el codigo modificado en un archivo aparte.

4.6 Modifique el codigo utilizando otra estructura de loop diferente a la utilizada en el ejemplo 4.1.

Ejemplo 4.2

Copie el siguiente fragmento en un archivo con extension “.py”, p.ej. ejemplo2.py
Trate de correrlo y verifique que no haya errores.

```
# Ejemplo de Programa con OPENGL
# ejemplo2.py
```

```
#las siguientes 3 lineas sirven para importar
#todas las funciones de los modulos GL, GLU y GLUT
```

```
from OpenGL.GLUT import *
from OpenGL.GL import *
from OpenGL.GLU import *
# la siguiente linea provee las herramientas para el manejo de graficas
import sys
```

#Las primeras letras de los comandos indican a que modulo pertenecen ej: gl, glu, glut

#def - indica el inicio de un nuevo bloque de codigo

```
def draw():  
    glClear(GL_COLOR_BUFFER_BIT)  
    glutWireTeapot(0.5)  
    glFlush() # Presenta el dibujo en la pantalla
```

#Presentaremos Graficos en estilo GLUT

#GLUT - GL Utilities Toolkit

#Provee los calculos para manejar multiples plataformas

#Windows, Mac, Linux

#sys.argv - permite ingresar parametros por consola,

si fuera necesario

```
glutInit(sys.argv)
```

#Utilizaremos un solo buffer de display - SINGLE

Y Utilizaremos el esquema de color RGB

```
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB)
```

```
glutInitWindowSize(500, 500)
```

```
glutInitWindowPosition(100, 100)
```

```
glutCreateWindow("Mi primer Programa OPENGL")
```

#indicamos a GLUT donde encontramos la funcion

#que crea la escena Grafica en este caso - draw

```
glutDisplayFunc(draw)
```

#la siguiente funcion permite correr el programa

#en un loop continuo. Permite a GLUT chequear

#si existen eventos de mouse o teclado

```
glutMainLoop()
```

End of program

4.7 Modifique el tamaño de la ventana y su posicion inicial

4.8. Cambie el tamaño de la tetera pruebe valores mayores y menores

4.9. Comente la linea `glutWireTeapot` y reemplacela con `glutWireSphere(0.5, 10, 10)`

4.10 Cambie a `glutWireSphere(0.75, 25, 25)` que es lo que cambia con cada parámetro?

4.11 Pruebe los comandos `glutWireCube(1.0)`

Ejemplo 4.3

Copie el siguiente fragmento en un archivo con extension “.py”, p.ej. ejemplo3.py
Trate de correrlo y verifique que no haya errores. Grabe las modificaciones propuestas en archivos diferentes.

```
#ejemplo3.py
#Crear un sistema de coordenadas con un punto de origen

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 plotpoints():
    #la siguiente funcion limpia la pantalla y pinta el fondo
    #definido en init
    glClear(GL_COLOR_BUFFER_BIT)
    #Definimos el color de ploteo a rojo - 3f indica usar 3 parametros
    #tipo punto flotante de 0.0 a 1.0
    glColor3f(1.0, 0.0, 0.0)
    #Informamos al sistema de graficos que vamos a dibujar puntos
    glBegin(GL_POINTS)
    #Definimos un punto en el centro (0.0,0.0)
    #2f indica que usaremos 2 numeros en formato punto flotante
    glVertex2f(0.0, 0.0)
    #Levantamos el lapiz - dejamos de dibujar
    glEnd()
    #presentamos la escena en pantalla
    glFlush()

#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("Dibujar Puntos")
    glutDisplayFunc(plotpoints)
```

```
init()  
glutMainLoop()
```

```
main()  
# End of Program
```

4.12 Cree dos colores de fondo uno “placentero a la vista” y otro “repugnante”

4.13 Cambie el color del pixel a blanco

4.14 Agregue la línea `glPointSize(2.0)` antes de `glBegin(GL_POINTS)` y experimente con el parámetro – Que sucede?

4.15 Dibuje varios puntos de diferentes colores y tamaños en cada cuadrante.

4.16 Comente el comando `glPointSize(2.0)` del punto 4.14 y cambie `glBegin(GL_POINTS)` por `glBegin(GL_LINES)` y dibuje dos puntos diferentes. Que sucede?

4.17 Dibuje un triángulo.

4.18 Reemplace `GL_LINES` con `GL_LINE_STRIP` y dibuje los puntos (0.0, 0.0), (1.0, 1.0), and (-1.0, 1.0). Que sucede?

4.19 Ahora utilice `GL_LINE_LOOP` con los mismos puntos.

4.20. Utilice el comando `glLineWidth(3.0)` debajo del comando `glPointSize(2.0)` y utilícelo con cualquiera de los comandos `GL_LINES`, `GL_LINE_STRIP`, o `GL_LINE_LOOP`

4.21 Dibuje un triángulo equilátero con un color diferente para cada lado

4.22 Dibuje los ejes X e Y de un sistema de coordenadas.

4.23 Dibuje 2 puntos simétricamente reflejados sobre y debajo del eje X

4.24 Dibuje 2 puntos reflejados a los lados de la línea $x=y$

4.25 Dibuje 2 triángulos reflejados a los lados del eje Y

Haga una captura de pantalla de su programa corriendo, y genere un archivo comprimido con el nombre : <apellido-nombre-lab01.zip> que incluya la captura de pantalla y el código final del programa ejemplo3.py.

Subalo al sitio web del curso.