

# CURSO: CC322 - 2017 -2

## Practica Lab 12

### 3. DESARROLLO

1. Verifique si tiene instalado un compilador de código en C (cpp, g++)  
Sino, proceda a instalarlo en su PC
2. Verifique si tiene un editor de texto adecuado para editar programas en C  
(p.ej. Geany, SublimeText, u otro con el que esté familiarizado) si no está – Proceda a Instalarlo.
3. Verifique si tiene instaladas las librerías OPENGGL Y GLUT. Si no están proceda a instalarlas.  
sudo apt-get install freeglut3-dev
4. Al final entregara un archivo con el nombre CC322\_Lab12\_<Nombre\_apellido>.zip con los archivos generados en la practica.

### 4 . Desarrollo

1. Descargue el programa esferas.c del repositorio del curso:

```
#include <GL/glut.h> // Header File For The GLut Library
#include <math.h>
/* Initialize z-buffer, projection matrix, light source,
 * and lighting model. Do not specify a material property here.
 */

int on=1;
GLfloat x=0.0, y=0.0, z=0.0;
void init(void)
{
    GLfloat ambient[] = { 0.0, 0.0, 0.0, 1.0 };
    GLfloat diffuse[] = { 1.0, 1.0, 1.0, 1.0 };
    GLfloat specular[] = { 1.0, 1.0, 1.0, 1.0 };
    GLfloat position[] = { 0.0, 3.0, 2.0, 0.0 };
    GLfloat lmodel_ambient[] = { 0.4, 0.4, 0.4, 1.0 };
    GLfloat local_view[] = { 0.0 };
    glClearColor(0.0, 0.1, 0.1, 0.0);
    glEnable(GL_DEPTH_TEST);
    glShadeModel(GL_SMOOTH);
    glLightfv(GL_LIGHT0, GL_AMBIENT, ambient);
    glLightfv(GL_LIGHT0, GL_POSITION, position);
    glLightModelfv(GL_LIGHT_MODEL_AMBIENT, lmodel_ambient);
    glLightModelfv(GL_LIGHT_MODEL_LOCAL_VIEWER, local_view);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
}
/* Draw twelve spheres in 3 rows with 4 columns.
 * The spheres in the first row have materials with no ambient reflection.
 * The second row has materials with significant ambient reflection.
 * The third row has materials with colored ambient reflection.
 *
 * The first column has materials with blue, diffuse reflection only.
 * The second column has blue diffuse reflection, as well as specular
 * reflection with a low shininess exponent.
 * The third column has blue diffuse reflection, as well as specular
 * reflection with a high shininess exponent (a more concentrated highlight).
 * The fourth column has materials which also include an emissive component.
 *
 */
void display(void)
{
    GLfloat no_mat[] = { 0.0, 0.0, 0.0, 1.0 };
    GLfloat mat_ambient[] = { 0.7, 0.7, 0.7, 1.0 };
    GLfloat mat_ambient_color[] = { 0.9, 0.2, 0.2, 1.0 };
    GLfloat mat_diffuse[] = { 0.1, 0.5, 0.8, 1.0 };
    GLfloat mat_specular[] = { 1.0, 1.0, 1.0, 1.0 };
    GLfloat no_shininess[] = { 0.0 };
    GLfloat low_shininess[] = { 5.0 };
    GLfloat high_shininess[] = { 100.0 };
    GLfloat mat_emission[] = { 0.1, 0.3, 0.2, 0.0 };
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    /* draw sphere in first row, first column
     * diffuse reflection only; no ambient or specular
     */
    glPushMatrix();
    glTranslatef (-3.75, 3.0, 0.0);
    glMaterialfv(GL_FRONT, GL_AMBIENT, no_mat);
    glutSolidSphere(1.0, 16, 16);
```

```

    glPopMatrix();
    /* draw sphere in first row, second column
    * diffuse and specular reflection; low shininess; no ambient
    */

    /* draw sphere in first row, third column
    * diffuse and specular reflection; high shininess; no ambient
    */

    /* draw sphere in first row, fourth column
    * diffuse reflection; emission; no ambient or specular reflection
    */

    /* draw sphere in second row, first column
    * ambient and diffuse reflection; no specular
    */

    /* draw sphere in second row, second column
    * ambient, diffuse and specular reflection; low shininess
    */

    /* draw sphere in second row, third column
    * ambient, diffuse and specular reflection; high shininess
    */

    /* draw sphere in second row, fourth column
    * ambient and diffuse reflection; emission; no specular
    */

    /* draw sphere in third row, first column
    * colored ambient and diffuse reflection; no specular
    */
    /* draw sphere in third row, second column
    * colored ambient, diffuse and specular reflection; low shininess
    */
    /* draw sphere in third row, third column
    * colored ambient, diffuse and specular reflection; high shininess
    */
    /* draw sphere in third row, fourth column
    * colored ambient and diffuse reflection; emission; no specular
    */
    glPushMatrix();
    glTranslatef (3.75, -3.0, 0.0);
    glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient_color);
    glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse);
    glMaterialfv(GL_FRONT, GL_SPECULAR, no_mat);
    glMaterialfv(GL_FRONT, GL_SHININESS, no_shininess);
    glMaterialfv(GL_FRONT, GL_EMISSION, mat_emission);
    glutSolidSphere(1.0, 16, 16);
    glPopMatrix();
    glFlush();
}

void reshape(int w, int h)
{
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    if (w <= (h * 2))
        glOrtho (-6.0, 6.0, -3.0*((GLfloat)h*2)/(GLfloat)w,
            3.0*((GLfloat)h*2)/(GLfloat)w, -10.0, 10.0);
    else
        glOrtho (-6.0*((GLfloat)w)/((GLfloat)h*2),
            6.0*((GLfloat)w)/((GLfloat)h*2), -3.0, 3.0, -10.0, 10.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void keyboard(unsigned char key, int a, int b)
{
    switch (key) {
        case 27:
            exit(0);
            break;
        case 'x':
            {
                x+=1.0; init(); glutPostRedisplay();
            } break;
        case 'o':
            {
                if (on) { glDisable(GL_LIGHT0); on=0; }
                else { glEnable(GL_LIGHT0); on=1; } //in this case I have to refer to init() functio
                // glEnable() doesn't enable specular reflection of material
                glutPostRedisplay();
            } break;
    }
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize (600, 450);
    glutCreateWindow(argv[0]);
    init();
    glutReshapeFunc(reshape);
    glutDisplayFunc(display);
    glutKeyboardFunc (keyboard);
    glutMainLoop();
    return 0;
}

```

- a) Modifique el programa para mostrar 12 esferas en un array de 3 filas y 4 columnas.
- b) Modifique la fuente de luz GL\_LIGHT0 para que muestre ademas de la propiedad "ambient" las propiedades "diffuse" y "position"
- c) Cambie el material de las esferas de modo que cada esfera sea diferente y con propiedades unicas :
  - Las esferas en la primera fila tienen materiales sin reflexion ambiental
  - Las esferas de la segunda fila tienen materiales con reflexion ambiental significativa
  - Las esferas de la tercera fila tienen materiales con reflexion ambiental coloreada.
  - La primera columna tiene materiales con una reflexion difusa azul
  - La segunda columna tiene materiales con una reflexion difusa azul, y tambien con reflexion especular y un nivel de brillantez alto.
  - La cuarta columna tiene materiales que incluyen tambien una componente de emision de luz
- d) Ampliar el programa para controlar dinámicamente la posición de la luz en 3 direcciones, y la dirección de la vista utilizando la entrada del teclado.  
Ponga los comentarios pertinentes en su código. Grabe el programa resultante.

2. Haga una copia del el programa anterior y modifiquelo dejando solamente la primera y la tercera fila de esferas.

- a) Haga que las esferas de la fila inferior sean 1/4 del diametro original, se encuentren sobre un "piso" .
- c) Utilizando el teclado para iniciar una "simulacion en 2D", muestre el movimiento de todas las esferas cuando 2 de las esferas de la fila superior caen y chocan con las de abajo (puede mover un poco las esferas de abajo para modificar el movimiento resultante).

Ponga los comentarios pertinentes en su código. Grabe el programa resultante