



```

/* program main loop */
while (!bQuit)
{
    /* check for messages */
    if (PeekMessage (&msg, NULL, 0, 0, PM_REMOVE))
    {
        /* handle or dispatch messages */
        if (msg.message == WM_QUIT)
        {
            bQuit = TRUE;
        }
        else
        {
            TranslateMessage (&msg);
            DispatchMessage (&msg);
        }
    }
}

```

Quitar

```

glClearColor (0.0f, 0.0f, 0.0f, 0.0f);
glClear (GL_COLOR_BUFFER_BIT);

glPushMatrix ();
glRotatef (theta, 0.0f, 0.0f, 1.0f);
glBegin (GL_TRIANGLES);
glColor3f (1.0f, 0.0f, 0.0f);    glVertex2f (0.0f, 1.0f);
glColor3f (0.0f, 1.0f, 0.0f);    glVertex2f (0.87f, -0.
glColor3f (0.0f, 0.0f, 1.0f);    glVertex2f (-0.87f, -0
glEnd ();
glPopMatrix ();

SwapBuffers (hDC);

```

```

    theta += 1.0f;
    Sleep (1);
}

```

Quitar

/\* shutdown OpenGL \*/

pasar  
Sleep(2000);  
aquí

- 5) Escriba Sleep(2000); después de SwapBuffers y quitar el comando glRotatef.

El pedazo de código arreglado quedaría así:

```

/* OpenGL animation code goes here */

glClearColor (0.0f, 0.0f, 0.0f, 0.0f);
glClear (GL_COLOR_BUFFER_BIT);

glPushMatrix ();
glBegin (GL_TRIANGLES);
glColor3f (1.0f, 0.0f, 0.0f);    glVertex2f (0.0f, 1.0f);
glColor3f (0.0f, 1.0f, 0.0f);    glVertex2f (0.87f, -0.5f);
glColor3f (0.0f, 0.0f, 1.0f);    glVertex2f (-0.87f, -0.5f);
glEnd ();
glPopMatrix ();

SwapBuffers (hDC);
Sleep (2000);

/* shutdown OpenGL */

```

Ahora vamos a Empezar a trabajar. Corre nuevamente el programa debe aparecer un triangulo con bordes de colores.

Aprendiendo comandos

**glColor3f** sirve para ponerle color a las líneas que dibujemos.

**Rojo** glColor3f(1,0,0); **Verde** glColor3f(0,1,0); **Azul** glColor3f(0,0,1);

Las líneas las dibujamos con vértices: glVertex2f(0,0); a glVertex2f(1,1);

Para decirle que trabajaremos con líneas debemos trabajar dentro del par de comandos:

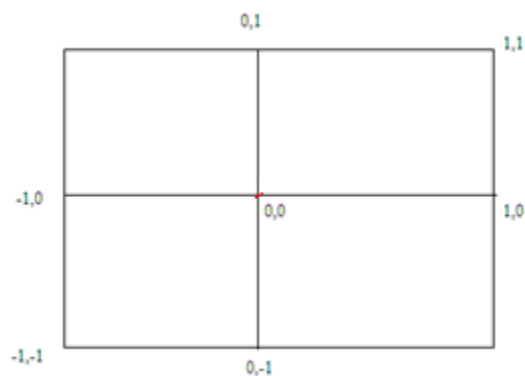
glBegin y glEnd.

El comando glBegin puede contener varios los parámetros, para este curso solo veremos uno

GL\_LINES

Crea una línea con dos vértices. glBegin(GL\_LINES);

**Espacio de trabajo:** está entre 0 y 1.



Veamos el ejemplo, cópialo y correlo en tu programa.

```
/* OpenGL animation code goes here */

glClearColor (0.0f, 0.0f, 0.0f, 0.0f);
glClear (GL_COLOR_BUFFER_BIT);

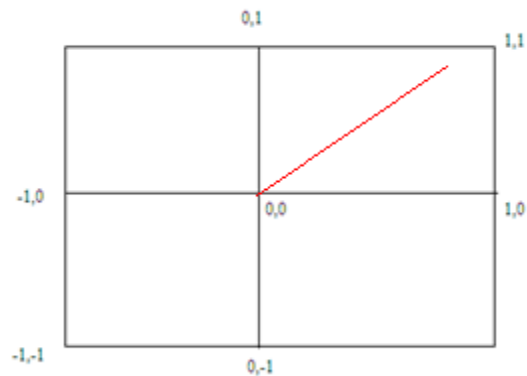
glPushMatrix ();
glBegin (GL_LINES);
glColor3f (1.0f, 0.0f, 0.0f);
glVertex2f (0.0, 0.0);
glVertex2f (0.9, 0.9); //se queda con el color de arriba:rojo

glEnd ();
glPopMatrix ();

SwapBuffers (hDC);
Sleep (2000);

/* shutdown OpenGL */
```

Nota que no necesitas asignar otro color más ya que con el rojo se ve la línea roja.



Deberás planear tus líneas en papel antes de escribirlas

Si deseo realizar una línea hacia arriba o una línea hacia abajo, coloco un if y declaro una variable x (al inicio del programa se deberá declarar int x;)

Corre este programa

```

x=0;

glClearColor (0.0f, 0.0f, 0.0f, 0.0f);
glClear (GL_COLOR_BUFFER_BIT);
glPushMatrix ();
glBegin (GL_LINES);
glColor3f (1.0f, 0.0f, 0.0f);
if (x==1)
{
    glVertex2f (0.0, 0.0);
    glVertex2f (0.9, 0.9);
}
if (x==0)
{
    glVertex2f (0.0, 0.0);
    glVertex2f (-0.9, -0.9);
}
glEnd ();
glPopMatrix ();

SwapBuffers (hDC);
Sleep (2000);

```

Dibuja una caja. Inténtalo y luego compáralo con el código. Si x=1 dibuja la caja si x=0, dibuja la línea hecha anteriormente.

```

x=1; // aqui cambiamos el valor para que aparezca lo
glClearColor (0.0f, 0.0f, 0.0f, 0.0f);
glClear (GL_COLOR_BUFFER_BIT);
glPushMatrix ();
glBegin (GL_LINES);
glColor3f (1.0f, 0.0f, 0.0f);
if (x==1) // dibujo caja
{
    glVertex2f (-0.3, -0.3);
    glVertex2f (0.3, -0.3);
    glVertex2f (0.3, 0.3);
    glVertex2f (0.3, 0.3);
    glVertex2f (-0.3, 0.3);
    glVertex2f (-0.3, 0.3);
    glVertex2f (-0.3, -0.3);
}
if (x==0) //dibujó línea
{
    glVertex2f (0.0, 0.0);
}

```

