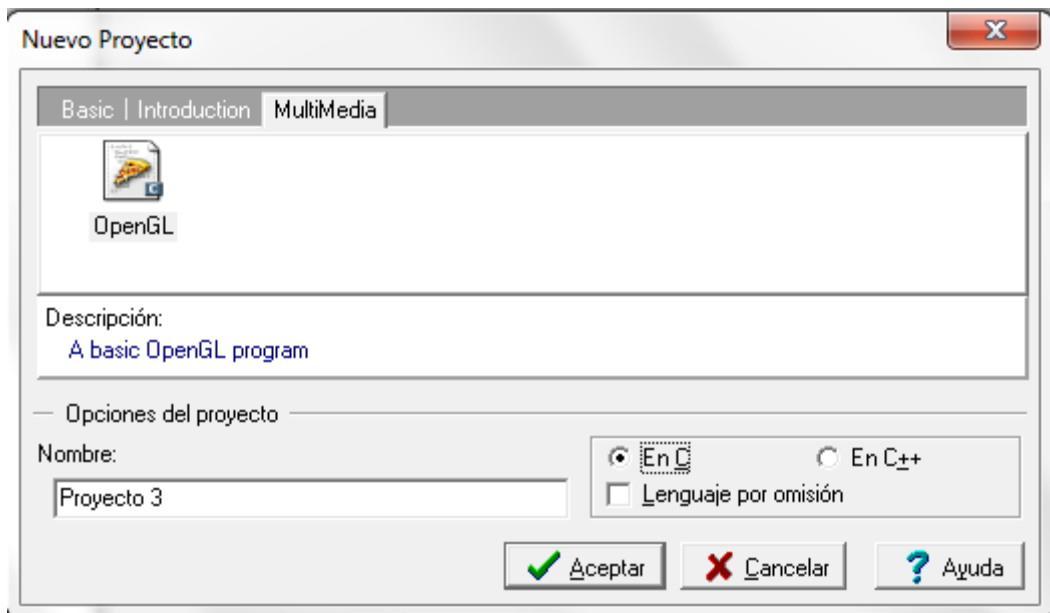


Programación Grafica en DEV C FCHE 2011

Objetivo1: Aprender a realizar dibujos en devc.

Objetivo2: Que el alumno se familiarice con el lenguaje y dibuje líneas, fomentando el gusto por la iniciación de programación.

Cree un nuevo proyecto en devc, escoja “multimedia” con c



Escribe el nombre del programa que desees y llámale como gustes al main de este programa.

Ya aparece código en el ejemplo.

Se deberá quitar desde que dice `/* program main loop */` hasta el segundo “else” y llave de inicio:

```

/* enable OpenGL for the window */
EnableOpenGL (hWnd, &hDC, &hRC);

/* 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);
        }
    }
    else
    {
        /* OpenGL animation code goes here */
    }
}

```

Y también las dos llaves finales

```

        SwapBuffers (hDC);

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

/* shutdown OpenGL */
DisableOpenGL (hWnd, hDC, hRC);

```

Quitamos también el theta y el arreglamos sleep(2000)

```

        glPopMatrix ();

        SwapBuffers (hDC);

        Sleep (2000);

/* shutdown OpenGL */

```

Guarda el programa, Vuelve a correr el programa, y no debe dar errores.(si da cierra todo y vuelve a empezar)

Si ya se ve un triangulo un rato y se quita Todo va funcionando bien

Solo trabajaremos con esta parte del código

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

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

SwapBuffers (hDC);

Sleep (2000);
```

El comando glRotatef, no lo utilizaremos en esta parte básica(sirve para rotar), debemos quitarlo.

```
/* 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);
```

EXPLICACION de estos comandos.

`glClearColor (0.0f, 0.0f, 0.0f, 0.0f);` sirve para cambiar el color del fondo

Contiene 4 argumentos 0.0f ,0.0f, 0.0f, 0.0f=> los tres primeros corresponden al color y el ultimo por el momento no se utilizara, siempre dejalo en cero.

El color puede descomponerse en colores primarios: R , G, B que sus siglas significan

r(red=rojo), G(Green=verde) y B(Blue=Azul)

Por lo tanto si queremos pantalla roja escribimos

```
glClearColor(1,0,0,0)
```

Pantalla verde `glClearColor(0,1,0,0)`

Pantalla Azul `glClearColor(0,0,1,0)`

Veamos nuestro código antes: correlo...observa tiene pantalla negra

Ahora edita solo este comando para que aparezca pantalla azul, asi:

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

glClearColor (0.0f, 0.0f, 1.0f, 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 (3000);
```

Siguiente comando

```
glClear (GL_COLOR_BUFFER_BIT);
```

Permite borrar el bufer de memoria. Deja como esta este comando después de asignar color a la pantalla.

Siguiente comando

Estos trabajan en par `glPushMatrix` y `glPopMatrix`, adentro siempre tendremos que trabajar

```

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.
glColor3f (0.0f, 0.0f, 1.0f);   glVertex2f (-0.87f, -0
glEnd ();
glPopMatrix ();

```

Siguiente comando

Estos también trabajan en par. glBegin con glEnd, dentro escribiremos nuestros vértices de líneas con un color.

```

glClearColor (0.0f, 0.0f, 1.0f, 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 (3000);

```

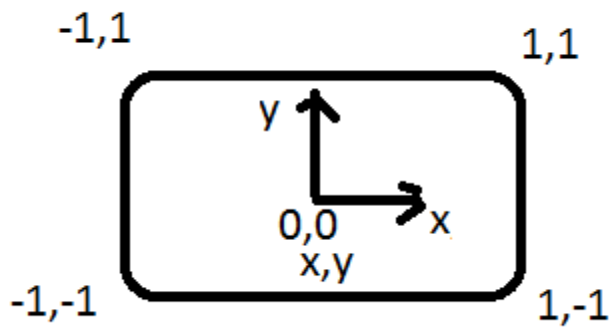
Comando glBegin(argumento) de forma básica utilizaremos el argumento para crear líneas únicamente.(existen otros para los que deseen profundizar :escribe en el internet glBegin y te aparecerán otras opciones y ejemplos)

Deberemos reescribirlo así: glBegin(GL_LINES)

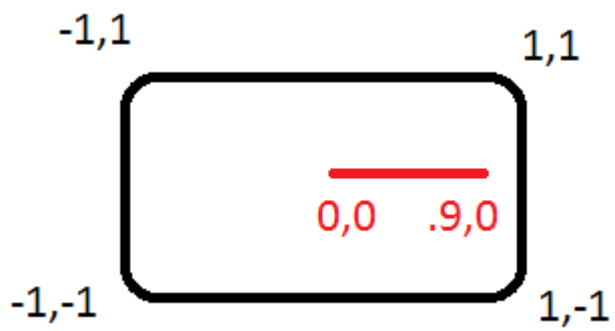
Después tenemos glColor3f (1.0f, 0.0f, 0.0f); que sirve para asignarle color a las líneas creadas por estos vértices (glvertex2f), los argumentos son iguales que el comando que cambia fondo de pantalla.

Y finalmente tenemos los vértices glVertex2f (0.0f, 1.0f); que su argumento contiene un x,y . este vértice es el punto de inicio de una línea y se coloca otro vértice como punto final para unir la línea.

Espacio de trabajo



Por lo tanto si queremos hacer una línea que empiece en cero y termine en horizontal el .9 se vería así



Insertemos los datos en nuestro programa, como se muestra a continuación, y después debemos correrlo.

```

/* OpenGL animation code goes here */

glClearColor (1.0f, 1.0f, 1.0f, 1.0f);
glClear (GL_COLOR_BUFFER_BIT);

glPushMatrix ();

glBegin (GL_LINES);
glColor3f (1, 0, 0);
glVertex2f (0, 0);
glVertex2f (.9, 0);

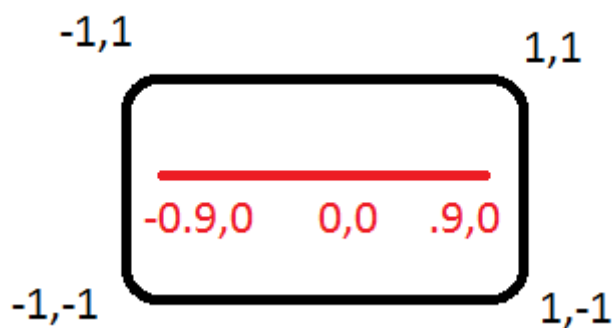
glEnd ();
glPopMatrix ();

SwapBuffers (hDC);

Sleep (2000);

```

Ahora una línea más larga, para que se vea así



Modifica el código siguiente y córrelo.

```

/* OpenGL animation code goes here */

glClearColor (1.0f, 1.0f, 1.0f, 1.0f);
glClear (GL_COLOR_BUFFER_BIT);

glPushMatrix ();

glBegin (GL_LINES);
glColor3f (1, 0, 0);
glVertex2f (-0.9, 0);
glVertex2f (.9, 0);

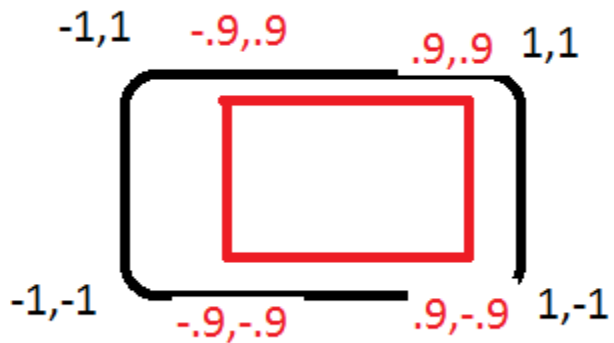
glEnd ();
glPopMatrix ();

SwapBuffers (hDC);

Sleep (2000);

```

A continuación vamos a dibujar un cuadrado



Por lo tanto contiene 4 líneas, 4 vértices. El vértice final de una línea es igual que el vértice inicial de la otra línea, por lo tanto deberemos escribir dos veces los vértices.

Copia las modificaciones y corre el programa

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

glClearColor (1.0f, 1.0f, 1.0f, 1.0f);
glClear (GL_COLOR_BUFFER_BIT);

glPushMatrix ();

glBegin (GL_LINES);
glColor3f (1, 0, 0);
glVertex2f (-0.9, -0.9); //vertice 1
glVertex2f (.9, -0.9);
glVertex2f (.9, -0.9);
glVertex2f (.9, .9);
glVertex2f (.9, .9);
glVertex2f (-.9, .9);
glVertex2f (-.9, .9);
glVertex2f (-.9, -.9);

glEnd ();
glPopMatrix ();

SwapBuffers (hDC);

Sleep (2000);
```

Ahora realiza un ejercicio, dibuja cualquier figura geométrica o caras (ojos nariz boca)

Sube tu tarea.

