Prácticas de Informática Gráfica - El entorno de trabajo

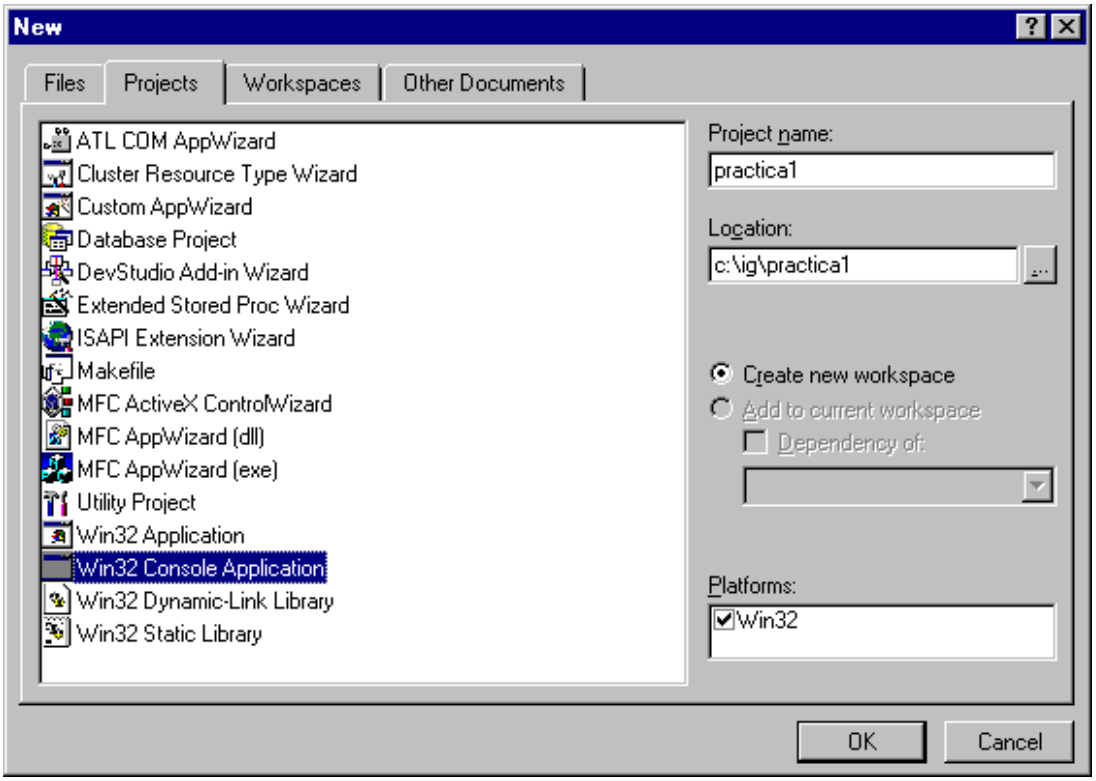
- **El entorno de desarrollo**

El desarrollo de programas está condicionado por la utilización de un determinado entorno de trabajo. El conocimiento del entorno de trabajo, junto con el lenguaje de programación, constituye la base de las tareas de programación. En este documento se muestran las características fundamentales del entorno de trabajo en el que puedes realizar las prácticas de Informática Gráfica, Microsoft Visual C++. Las prácticas las programarás en el lenguaje de programación C. La biblioteca de funciones gráficas que utilizarás es OpenGL de Silicon Graphics.

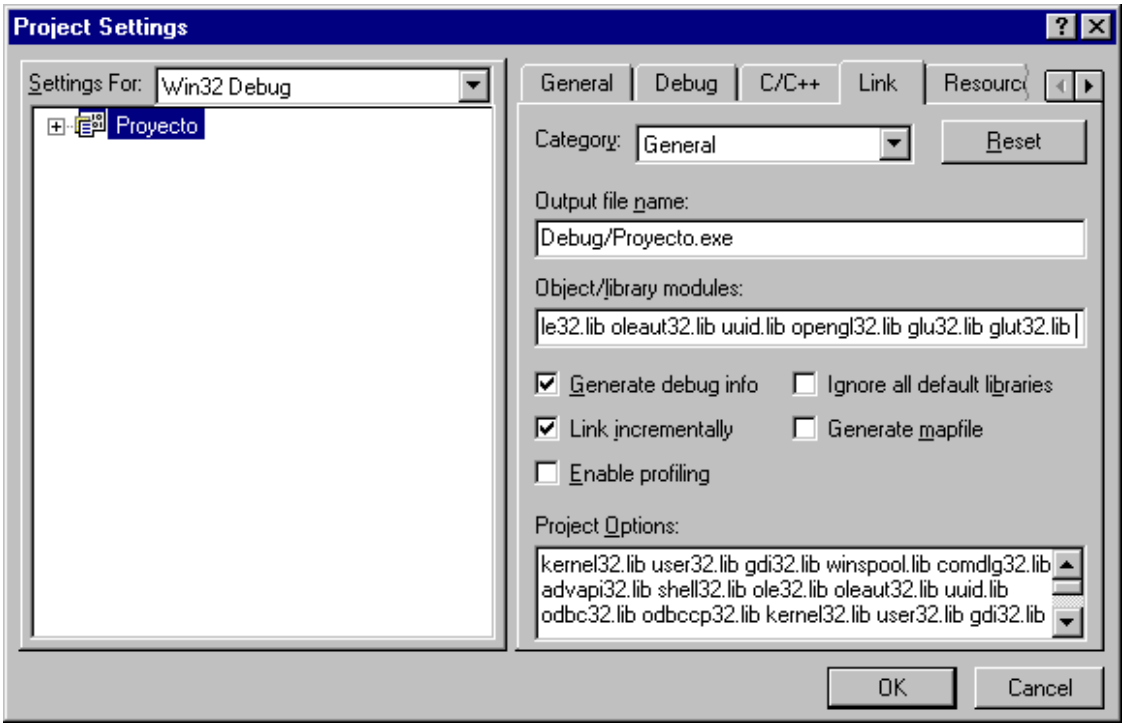
EL ENTORNO DE DESARROLLO

El objetivo de este documento es que te familiarices con el entorno de trabajo Microsoft Visual C++. Para empezar aprenderemos a crear un proyecto sencillo (aplicación) para, posteriormente, compilarlo y ejecutarlo. Los pasos que debes seguir en la creación de un proyecto son los siguientes:

- Inicia Visual C++: selecciona del menú de Inicio > Programas > Microsoft Visual C++ 6.0 > Microsoft Visual C++ 6.0
- Crea el proyecto: Opción File / New. El tipo del proyecto debe ser Win32 Console Application. Has de elegir una ubicación para el proyecto (por ejemplo, c:\ig) y darle un nombre (por ejemplo, practica1).



- A la pregunta "What kind of Console Application do you want to create?" responde con la opción "An empty project".
- Guarda los ficheros cuadrado.c y cuadrado.h en el directorio que acabas de crear para tu proyecto (c:\ig\practica1).
- En el siguiente paso añade al proyecto los ficheros que lo forman. Para ello utiliza la opción Project / Add to Project / Files. Los ficheros que añadirás son el fichero fuente cuadrado.c y el fichero de cabecera cuadrado.h, que previamente has guardado en tu directorio de proyecto.
- Por último has de indicar que bibliotecas de funciones quieres añadir a tu proyecto. Para ello, selecciona la opción de menú Project / Settings y en la ventana que se abrirá, selecciona la solapa Link. En la Categoría (Category) General (General) añade a Object/library modules las siguientes bibliotecas (ficheros con extensión .lib): opengl32.lib, glu32.lib y glut32.lib. Pulsa OK para completar el proceso.

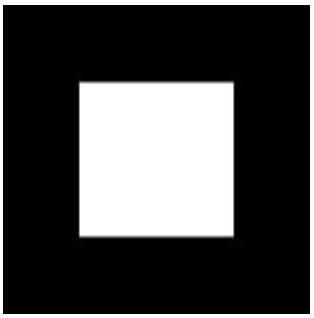


Una vez realizados los pasos anteriores ya puedes compilar el proyecto con la opción: *Build / Build practica1.exe*. Si todo funciona correctamente, en el área de mensajes debe aparecer los siguientes mensajes:

```
-------Configuration: practica1 - Win32 Debug------
        Compiling...
        cuadrado.c
        Linking...
        practica1.exe – 0 error(s), 0 warning(s)
```

Como puedes comprobar se ha creado un nuevo archivo *practica1.exe* el cual puedes ejecutar con la opción *Build / Execute practica1.exe*. El resultado es una ventana en la que aparece un cuadrado blanco sobre fondo negro, además de la consola de la aplicación.

Si el programa se ha ejecutado correctamente puedes cerrar la ventana OpenGL y la consola; para ello selecciona la consola y pulsa cualquier tecla.



Analiza la estructura del programa. La estructura de este programa es el modelo que seguirás para construir los programas que desarrollarás durante el resto de las prácticas.

Cuadrado.c

```
#include <gl/glut.h>
#include <stdio.h>
#include "cuadrado.h"

/*************************************************************************/

/*  Funcion de dibujado                                  */
/*  Parametros: Ninguno                                  */
/*  Salida: Ninguna                                      */

/*************************************************************************/

void Dibuja(void)
{
      /* Establece el color de borrado */
      glClearColor (0.0f, 0.0f, 0.0f, 0.0f);

      /* Borra el buffer de color */
      glClear (GL_COLOR_BUFFER_BIT);

      /* Establece el color de dibujo */
      glColor3f (1.0f, 1.0f, 1.0f);

      /* Crea un poligono 2D (cuadrado) */
      glBegin (GL_POLYGON);
            glVertex2f (-0.5f, -0.5f);
            glVertex2f (-0.5f, 0.5f);
            glVertex2f (0.5f, 0.5f);
            glVertex2f (0.5f, -0.5f);
      glEnd ();

      /* Se asegura de que se ejecutan todas las ordenes */
      glFlush ();
}

/*************************************************************************/

/* Establece el area visible                                  */
/* Parametros: int ancho --> Ancho del area visible           */
/*             int alto --> Alto del area visible             */
/* Salida: Ninguna                                            */

/*************************************************************************/
```

```
void TamanyoVentana (int ancho, int alto)
{
	glViewport (0, 0, ancho, alto);
}

/*****************************************************************************/

/* Inicia las propiedades de la vista                                        */
/* Parametros: Ninguno                                                       */
/* Salida: Ninguna                                                           */

/*****************************************************************************/

void IniciaVista (void)
{
	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);
}

/*****************************************************************************/

/* Abre una ventana OpenGL                                                   */
/* Parametros: int numeroArgumentos --> El numero de argumentos en la llamada al
programa                                                                     */
/*             char ** listaArgumentos --> Vector de cadenas con cada argumento */
/* Salida: Ninguna                                                           */

/*****************************************************************************/

void AbreVentana (int numeroArgumentos, char ** listaArgumentos)
{
	glutInit(&numeroArgumentos, listaArgumentos);
	glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
	glutInitWindowSize (500, 500);
	glutInitWindowPosition (100, 100);
	glutCreateWindow (listaArgumentos[0]);
	glutDisplayFunc (Dibuja);
	glutReshapeFunc (TamanyoVentana);
	IniciaVista ();
}

/*****************************************************************************/

/* Funcion principal                                                         */
/* Parametros: int numeroArgumentos --> El numero de argumentos en la llamada al
programa                                                                     */
```

```
/*            char ** listaArgumentos --> Vector de cadenas con cada argumento          */
/* Salida: Un entero que se devuelve al sistema al acabar la ejecucion del programa    */

/*****************************************************************************/

int main(int numArgumentos, char ** listaArgumentos)
{
      /* Crea la ventana de la aplicaci¢n */
      AbreVentana (numArgumentos, listaArgumentos);

      /* Establece el bucle principal de control de OpenGL */
      glutMainLoop();

      return (0);
}

/*****************************************************************************/
```

cuadrado.h

```
#ifndef OCUADRADO_H
#define OCUADRADO_H

/* Abre una ventana OpenGL */
void AbreVentana (int numeroArgumentos, char ** listaArgumentos);

/* Funcion de dibujado */
void Dibuja(void);

/* Establece el area visible */
void TamanyoVentana (int alto, int ancho);

/* Inicia las propiedades de la vista */
void IniciaVista (void);

#endif
```