El compilador BloodShed DEV-C++ 4

El DEV C++ es un buen entorno de desarrollo integrado (IDE) para el compilador Mingw (Todo ello freeware). Permite crear programas en C/C++ para DOS o Windows. Se distribuye bajo licencia GPL y por supuesto incluye el código fuente (en Delphi). Ha sido distribuido recientemente por numerosas revistas; además se puede encontrar en <u>http://www.bloodshed.net/</u>.

Su instalación se reduce a ejecutar el setup.exe y eligiendo la opción 'típica' estaremos preparados para usarlo en unos instantes.

Una vez abierto el programa vamos al menú Options. Primero pulsamos sobre Compiler Options:

Ĩ.	ry below to be searched for include files.
Add the followin	ng commands when calling compiler.
You can change th user can also add directories by sep	e directory paths of your GCC compiler system (Cygwin the paths to compile with it). You can add multiple arating them with a semi-colon ().
and the second states and	
Bin directory:	C\PROGRAMACIóN\DEVC++\Bin\
Bin directory: C include files:	C\PROGRAMACIoN\DEVC++\Bin\ C\PROGRAMACIoN\DEVC++\Include\
Bin directory C include files: C++ include files:	C\PROGRAMACIóN\DEVC++\Bin\ C\PROGRAMACIóN\DEVC++\Include\ C\PROGRAMACIóN\DEVC++\Include\G++;C\PROGR

Aquí debería estar todo autoconfigurado. Los directorios arriba señalados dependerán de dónde lo hayas instalado tú.

Luego pulsamos en Environment Options:

Auto-Arrange Do got show Project Explorer Do not open source files when	Make compile results stay on top Maximize editor when opening a tile opening a project
Auto-Saving: □ Create backup file ☑ Desktop position	Save toolbars availability Editor size & position
Default directory C \PROGRAMACIóN\	
Execution. Minimize during execution Execute in directory of executal	ble
Give the following parameters v	when executing a compiled project

Pon el default directory que quieras. Las demás solapas sólo afectan a la apariencia del entorno.

Veamos como configurarlo para compilar el ejemplo tut_2_0.zip. Supongamos que hemos descomprimido dicho ejemplo en c:\programación\prog.

Crearemos un nuevo proyecto en File --> New Project (o icono correspondiente):

roject Additio	onal GUI Toolk	its Graphics			
Windows Application	Console Application	WinMain() Project	DUL	Empty Project	
scription: As	tandard Window	s application.			
Make default	language			√ QK	X Cancel

Seleccionamos C project y Windows Application. Las demás solapas las dejamos sin tocar. Luego nos pedirá un nombre para nuestro proyecto y dónde crearlo.

Abrirá una ventana con una plantilla de aplicación Windows. Como no la necesitamos, vamos a project D & Remove from project D & untitled y elegimos no salvar.

Ahora incluiremos en el proyecto los ficheros de nuestro ejemplo. Para ello vamos a Project ∂Add to project→elegimos el directorio donde tengamos los ficheros (c:\programación\prog) y manteniendo pulsada la tecla de control pinchamos sobre todos los '.c'.

Ahora sólo falta especificar las propiedades de nuestro proyecto: Project De Project Options.

		Project type:	
	🕒 Load icon	Compile C++ project	
W -		Do not create a console	
	Con Library	Create a DLL	
Further obj options (se	ect files or linker parate filenames by		
-lopengl32 -lglu32		🙆 Load <u>o</u> bject files	
		AN A	
Extra comp	piler options (only for thi	s project) :	
Extra comp	ailer options (only for thi	s project) :	
Extra comp Extra includ	oiler options (only for thi de directories (separati	s project) : e paths by a semicolon ".") :	
Extra comp Extra incluc c:\program	oller options (only for thi de directories (separate nación\prog\src	s project) : e paths by a semicolon ".") :	
Extra comp Extra incluc c:\program Resource 1	oller options (only for thi de directories (separate nación\prog\src files (separate filename	s project) : e paths by a semicolon ".") : es by a semicolon ".") :	
Extra comp Extra includ c:\program Resource I C:\program	oller options (only for thi de directories (separate nación\prog\src files (separate filename mación\prog\rsrc.rc	s project) : e paths by a semicolon ".") : es by a semicolon ".") : Coad resource (.rc) file:	
Extra comp Extra incluc c:\prograr Resource I C:\prograr Change pri	oller options (only for thi de directories (separate nación\prog\src files (separate filename mación\prog\rsrc.rc oject name :	s project) : a paths by a semicolon ".") : is by a semicolon ".") : boad (esource (.rc) file:	
Extra comp Extra incluc c:\prograr Resource 1 C:\prograr Change pri 1ut2_0	oller options (only for thi de directories (separate nación\prog\src files (separate filename mación\prog\rsrc.rc oject name :	s project) : e paths by a semicolon ".") : is by a semicolon ".") : Load resource (.rc) file:	

Aquí podemos seleccionar el icono de nuestra aplicación y lo que es más importante: especificar el uso de las librerías opengl y glu. Para ello en 'further object files' escribimos "-lopengl32 -lglu32" (sin las comillas). Además en 'extra include directories' hemos de especificar el directorio de nuestro ejemplo para que el compilador pueda leer nuestros '.h'.

Bien, ahora ya estaríamos preparados para compilar: (Execute D & Compile). Pero para que el ejemplo funcione correctamente en Dev C++ es necesario añadir a 'sys_win32.c' las siguientes líneas (al comienzo del fichero):

#ifndef CDS_FULLSCREEN #define CDS_FULLSCREEN 4 #endif

Pues ya está ¡a compilar!