

# INTERACCIÓN PERSONA-MÁQUINA

## Ejercicios OpenGL

**Curso 2012-13**

### **Configuración de OpenGL en el entorno Dev-C++**

---

Para la realización tanto de estos ejercicios como de la práctica se concede libertad de elección del entorno de desarrollo, así como del sistema operativo, siempre y cuando se implementen en C/C++ y OpenGL. Si bien en esta guía se dan instrucciones concretas para su implementación en Dev-C++ para Windows.

Será necesario descargar de la web <http://orwelldevcpp.blogspot.com.es/> el instalador Dev-Cpp 5.4.1 MinGW 4.7.2 Setup.exe (las versiones pueden variar) y de Moodle los archivos `holamundo.cpp` y `librerias.zip`.

### **Instalación del entorno Dev-C++**



La instalación del ejecutable descargado no tiene ninguna dificultad.

### **Instalación de las librerías**

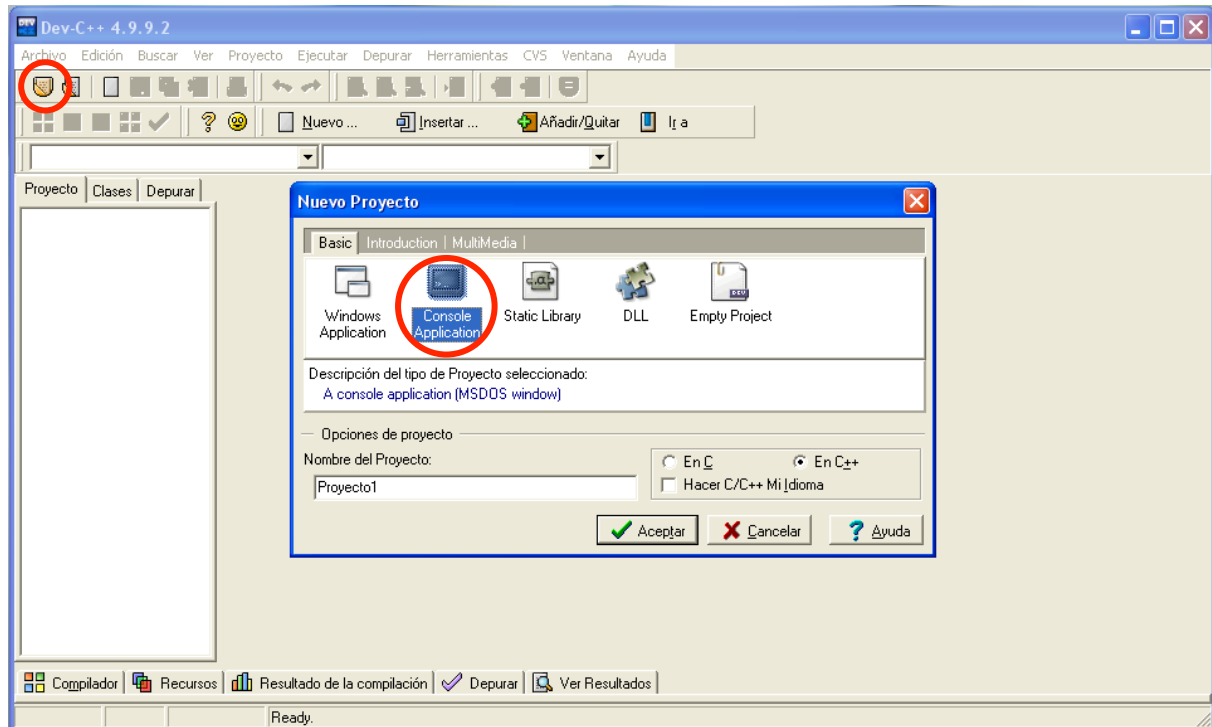
El archivo `librerias.zip` contiene las librerías necesarias para programar y compilar con GLUT y con otras las librerías de imágenes y sonido en Dev-C++. Los ficheros incluidos en el ZIP deben instalarse en diferentes carpetas según lo siguiente:

- El contenido de la carpeta `include` debe copiarse en la carpeta `C:\Archivos de programa\Dev-Cpp\MinGW32\include`.
- El contenido de la carpeta `lib` debe copiarse en la carpeta `C:\Archivos de programa\Dev-Cpp\MinGW32\lib`.
- El contenido de la carpeta `system32` debe copiarse en la carpeta `C:\WINDOWS\SYSTEM32`.

## Configuración de un nuevo proyecto

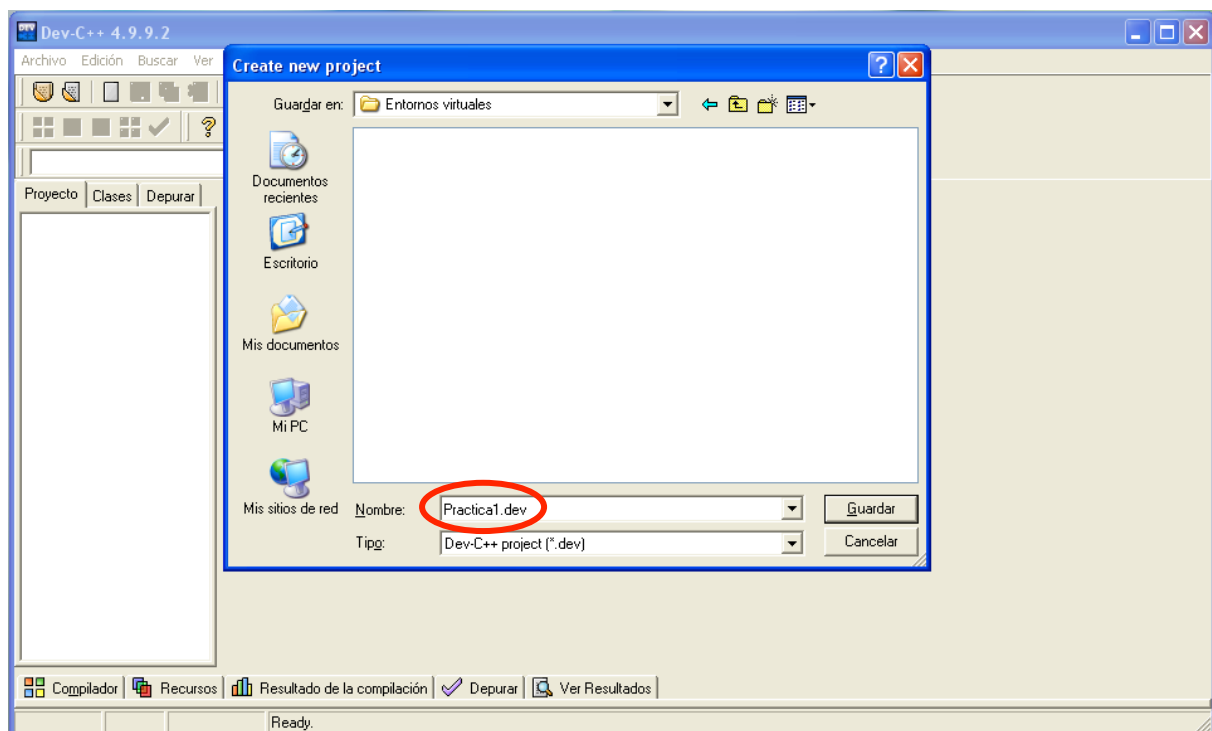
1. Crear un nuevo proyecto pulsando sobre el icono y elegir la aplicación *Console Application*.

En las opciones del proyecto marcar la opción *En C++*.

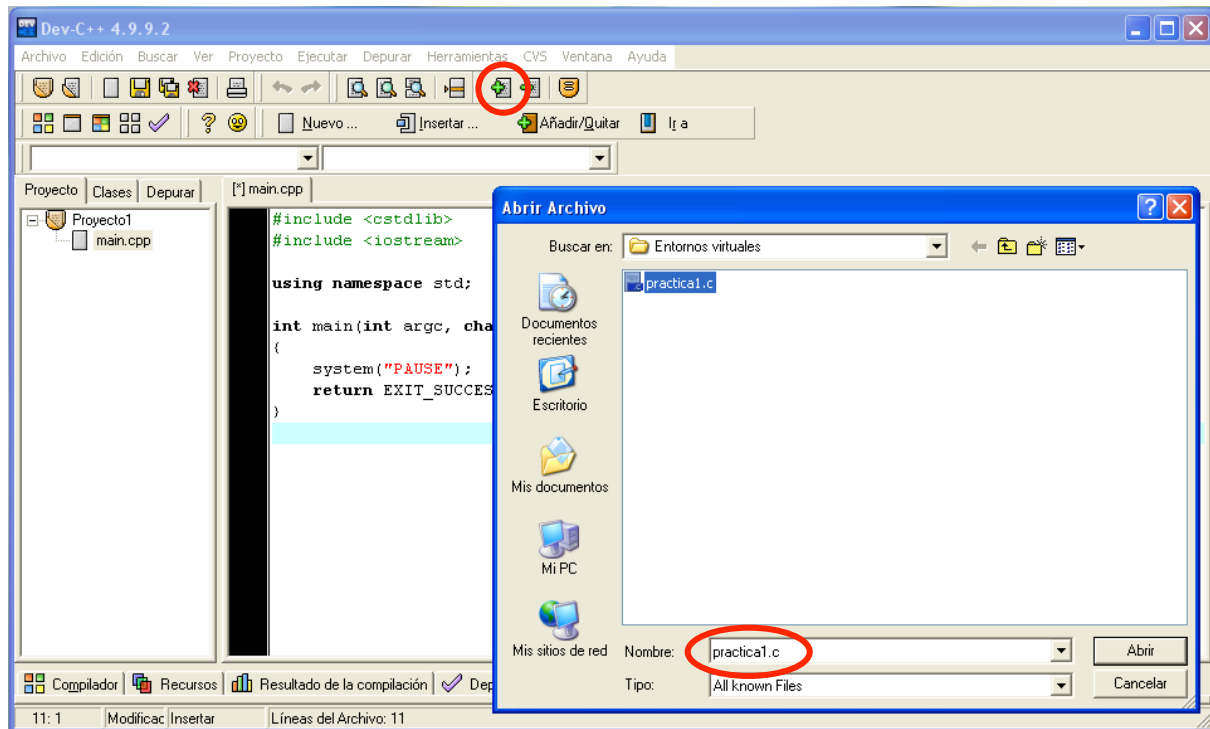


**ATENCIÓN: No seleccionar el proyecto OpenGL de la pestaña Multimedia.**

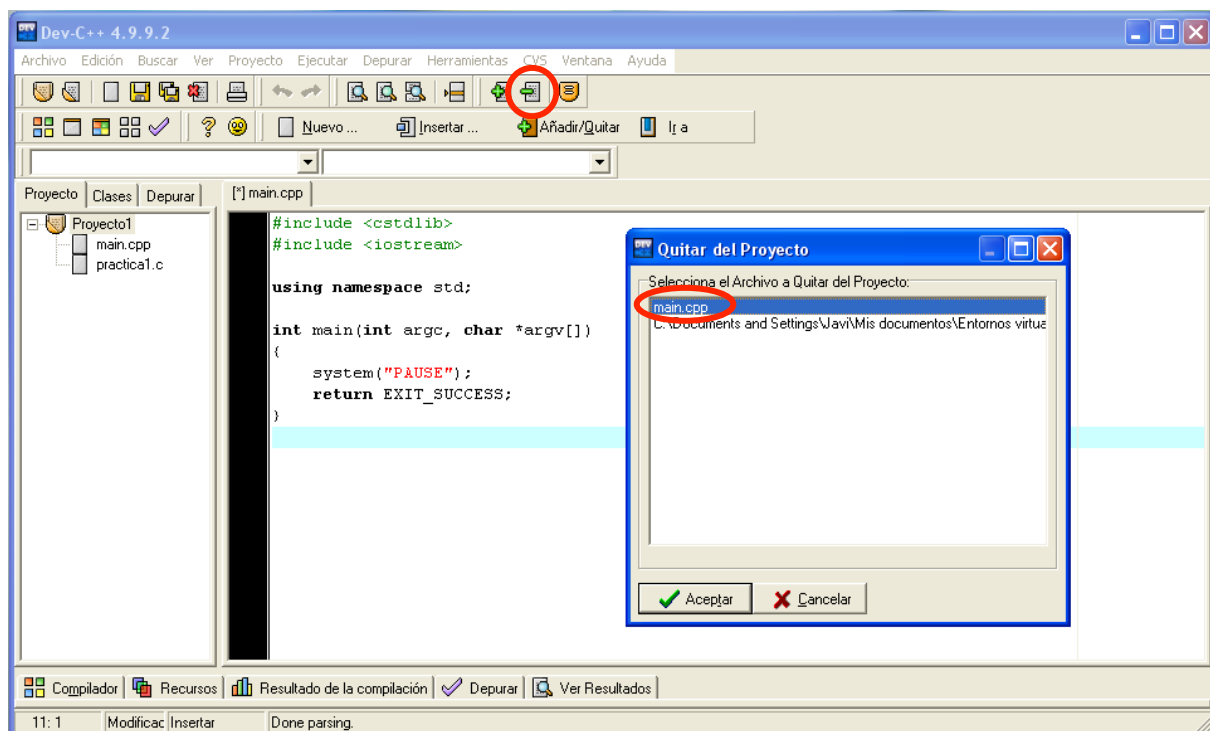
2. Grabar el proyecto dándole un nombre (p.e., holamundo.dev).



3. Automáticamente se crea un archivo `main.cpp` donde se puede escribir el código. También se puede sustituir por un archivo ya creado pulsando el icono con el signo más *Añadir a proyecto...* Para esta práctica, seleccionar `holamundo.cpp`.



A continuación hay que pulsar el icono con el signo menos *Quitar del proyecto...* para eliminar `main.cpp`.

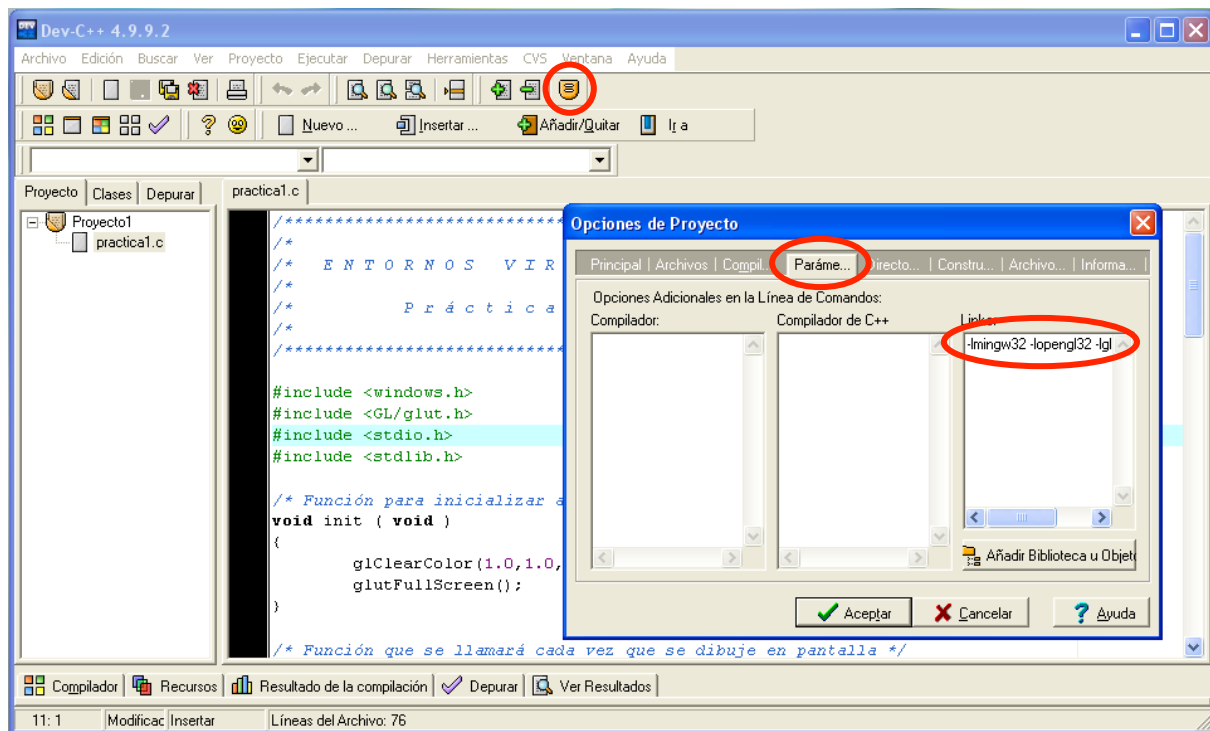


4. Por último, hay que configurar las opciones del proyecto indicándole al montador (linker) las librerías que hay que enlazar:

```
-lmingw32 -lopengl32 -lglu32 -lfreeglut
```

Si queremos utilizar la librería DevIL, para cargar imágenes, y la librería SDL\_mixer, para utilizar sonido, hay que añadir:

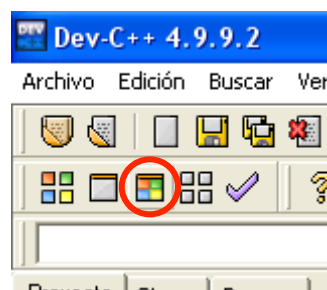
```
-lDevIL -lILU -lILUT -lSDLmain -lSDL -lSDL_mixer
```



Esta forma de crear aplicaciones con Dev-C++ nos abre una consola (ventana de MS-DOS) donde podemos mostrar trazas de ejecución del programa con la función `printf(...)`, independientemente de la información que se muestre con la ventana de OpenGL.

### Compilación y ejecución

Automáticamente se compila, se monta y se ejecuta pulsando el icono marcado:



La ejecución abre una ventana de MS-DOS y otra ventana con la aplicación OpenGL.

## **Ejercicio 1 – Hola Mundo OpenGL**

---

Crear un proyecto en Dev-C++ e incluir el archivo `holamundo.cpp` en el proyecto, compilarlo y ejecutarlo. Este código muestra un cuadrado rojo sobre un fondo blanco. Pulsando la **tecla F**, la aplicación se muestra a pantalla completa. Pulsando la **tecla W** se reduce la imagen a una ventana de Windows. Pulsando la tecla **Escape** se termina la ejecución. Comprobar que todo funciona correctamente.

A continuación, realizar las siguientes modificaciones y comprobar el efecto que causa cada una de ellas:

- Modificar el tamaño y posición de la ventana.
- Añadir una función de manejo del ratón que cierre la aplicación si se pulsa el botón izquierdo.
- Intercambiar la posición en el código de las 2 últimas funciones `glVertex2i` y ver qué pasa. Dejarlo como estaba.
- Cambiar el color del fondo.
- Cambiar el color de los vértices, poniendo un color distinto a cada uno de ellos. Dejarlo como estaba.
- Añadir un cuadrado verde con las mismas posiciones del rojo que aparezca en pantalla a la derecha. Definirlo en el código antes del cuadrado rojo.
- Poner una rotación de 45° antes del rectángulo verde.
- Rotar 45° el rectángulo verde sin que afecte al cuadrado rojo.

## **Ejercicio 2 – Movimiento de objetos**

---

Basándose en el código del ejercicio 1, crear otro proyecto en el que se realice lo siguiente:

- Ejecutar la aplicación siempre a pantalla completa.
- Colocar un rectángulo alargado cerca de la parte inferior de la pantalla. Con las pulsaciones de las teclas de cursor izquierda y derecha, se deberá desplazar una determinada cantidad de píxeles en el mismo sentido de la flecha, limitando el movimiento a los bordes de la pantalla.
- Colocar un polígono de 8 lados (octógono). Con la pulsación de la tecla espacio, iniciar un movimiento en diagonal del polígono, controlando las colisiones con los bordes de la pantalla.
- Dibujar una matriz de rectángulos en la parte superior de la pantalla.

Todo debe ser independiente del tamaño de la pantalla.