



# **Guion de prácticas**

## *Proyecto Final - Parte 1*



# **Metodología de la Programación**

Grado en Ingeniería Informática

Prof. David A. Pelta

## Introducción al guion

En este guion volveremos a trabajar con clases y compilación separada. Utilizaremos un módulo muy simple para generar gráficos en pantalla.

Esta práctica NO requiere entrega pero es un requisito indispensable para realizar el proyecto final de la asignatura.

### Paso 1

Descargue de PRADO el código de ejemplo dibujos.zip. Encontrará cuatro ficheros: `main.cpp`, `miniwin.cpp`, `miniwin.h` y `compila`. En este último puede ver las instrucciones necesarias para compilar el código desde la línea de comandos. Compílelo y ejecútelo.

Verá una ventana donde se muestran dos puntos enlazados, donde uno se mantiene fijo y el otro rota a su alrededor. Cada cierto tiempo, los puntos se intercambian. Para terminar el programa, tiene que pulsar la tecla “Esc”.

Abra el fichero `main.cpp`. Verá que ha ejecutado el ejemplo 1. Comente el código asociado a dicho ejemplo, y quite los comentarios del ejemplo 2. Vuelva a compilar y ejecutar. Ahora verá un triángulo que rota alrededor de alguno de sus vértices.

Para la visualización se utiliza un módulo muy simple llamado MiniWin. Toda la documentación de la misma está disponible en la página <http://miniwin.readthedocs.io/en/latest/>. En `miniwin.h` puede ver las funciones disponibles.

Un elemento importante a tener en cuenta es como se definen las coordenadas en MiniWin. La Figura 1 permite observar que el punto (0,0) se encuentra en el extremo superior izquierdo de la pantalla.

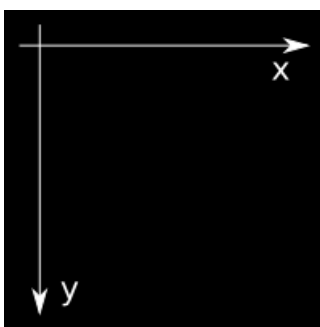


Figura 1: Sistema de coordenadas en MiniWin

Estudie el código y luego haga cambios para pintar otras figuras, compruebe el efecto de eliminar la instrucción `borra()`, etc.

Cree un proyecto en NetBeans siguiendo los pasos usuales. Para que se pueda compilar, acceda a las propiedades del proyecto, y luego seleccione `linker`. Verá una ventana similar a la mostrada en la Fig. 2(a). En el apartado `Additional Options`, pinche en el cuadrado de la derecha y se abrirá una nueva ventana como la que se muestra en la Fig. 2(b). Escriba el texto `-pthread -lX11` en el apartado correspondiente.

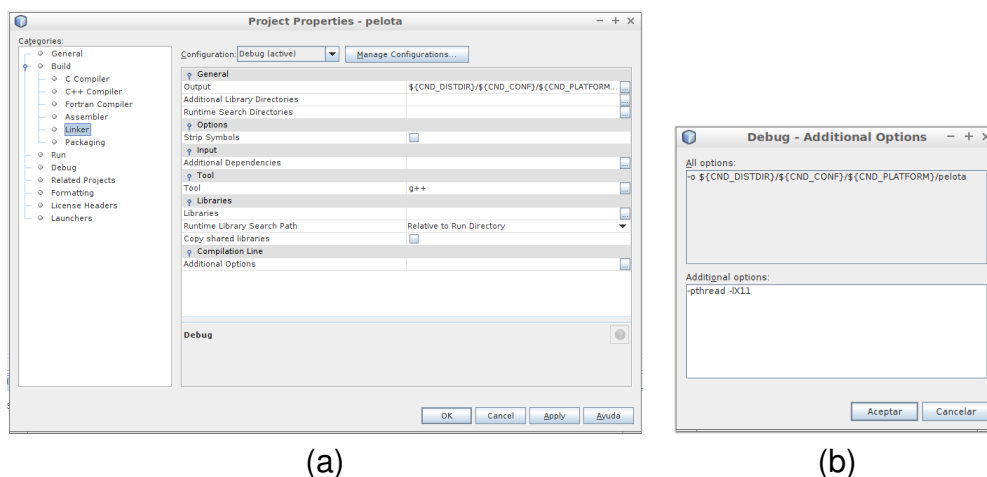


Figura 2: Ajustes de las propiedades del proyecto.

## Paso 2: Clase Punto2D

En lugar de utilizar el struct `Punto`, se pide diseñar una clase `Punto2D` (análoga a `Punto`) que contenga simplemente la posición en el plano representada como dos valores  $x, y$  de tipo `float`.

Respecto a los métodos, y además de los `set/get`, la clase debe proveer:

- `Punto2D()`: Constructor por defecto. Inicializa la posición con valores aleatorios. Supondremos que  $0 \leq x \leq MIN\_X$  y  $0 \leq y \leq MIN\_Y$ . Las constantes `MIN_X`, `MIN_Y` hacen referencia al tamaño mínimo que tendrá la ventana de visualización. Puede fijar ambas en 800.
- `Punto2D(float x, float y)`: Constructor con parámetros. Asigna la posición inicial al objeto.
- `Distancia(const Punto2D & otro)`: Método para calcular la distancia entre dos puntos.
- `Rotar(const Punto2D & centro, double rads)`: Método para rotar un punto alrededor de otro. La rotación estará expresada en radianes. Si quiere transformar grados a radianes debe hacer `rads = grados * (M_PI / 180.0)`. La constante `M_PI` está definida en `cmath`.
- `Mover(float dx, float dy)`: Método para mover un punto. Recibe los desplazamientos para  $x$  e  $y$ .

Recupere los primeros ejemplos de compilación separada para recordar que debe incluir en los ficheros `.cpp` y `.h`.

## Paso 3: Módulo pintar

Debe notar que la Clase `Punto2D` “no sabe nada” de Miniwin y por tanto, no tiene sentido que la clase provea métodos para “dibujar” un objeto.

Es por ello que se propone implementar las funciones dedicadas a pintar un objeto de la clase Punto2D o una línea entre dos objetos, en un módulo separado que se llamará `pintar`.

Posteriormente, adapte el programa de prueba para que utilice objetos de la Clase Punto2D en lugar de structs.

## Paso 4

Implemente un programa para que se muestren  $N$  puntos generados al azar que rotarán sobre el centro de la ventana. Para ello, utilice un array dinámico `nube` de tipo Punto2D, en una ventana de tamaño 800 por 800. Es decir, el punto de rotación será el (400, 400).

Agregue al módulo `pintar` sendas funciones que permitan pintar y rotar todos los puntos del array.

## Importante

Estas instrucciones deben complementarse con las indicaciones dadas durante las sesiones de práctica.