

---

# PRÁCTICA 7.b

## Mejora del “Paint Básico 2D”

---

### ■ Descripción del ejercicio

El objetivo de esta práctica es mejorar el “Paint Básico 2D” realizado en la práctica 7 introduciendo las siguientes nuevas funcionalidades:

- Cada figura tendrá sus propios atributos.
- Nuevas formas de dibujo
- Nuevos atributos de dibujo

La aplicación permitirá dibujar diferentes formas geométricas (cuadrados, rectángulos, etc.) con diferentes atributos (color, grosor, etc.). Para ello, se incorporará una barra de herramientas que dé acceso a todos los elementos necesarios para poder dibujar, incluyendo formas y atributos. El usuario podrá dibujar utilizando la forma y atributos seleccionados (véanse siguientes subsecciones), considerando los siguientes requisitos de carácter general:

- El lienzo mantendrá todas las figuras que se vayan dibujando.
- Cada figura tendrá sus **propios atributos**<sup>1</sup>, independientes del resto de formas (es decir, no compartirán los mismos valores). Cuando se dibuje la forma por primera vez, ésta usará los atributos que estén activos en ese momento (que no tienen por qué coincidir con los de las figuras que ya estén en el lienzo).
- El usuario podrá **seleccionar las figuras** ya dibujadas y **editar sus atributos**, esto es, se podrá modificar cualquiera de sus propiedades (color, trazo, relleno, etc.) sin más que cambiarla en la barra de herramientas (y los cambios se tendrán que ver reflejados en la forma). Además, el usuario podrá **mover la figura** seleccionada.

### ● Formas de dibujo

Además de las formas introducidas en la práctica 7:

- Línea recta
- Rectángulo
- Elipse

se aconseja incorporar nuevas formas geométricas<sup>2</sup>:

- Rectángulo redondeado
- Arco
- Curva con un punto de control
- Curva con dos puntos de control
- Trazo libre
- Polígono
- Una forma personalizada (área) que defina una nueva figura

---

<sup>1</sup> Nótese que, a diferencia de la práctica 7, en la que todas las formas se mostraban con los mismos atributos, en este caso cada forma tendrá asociado un conjunto de atributos propio.

<sup>2</sup> Nótese que de relación de formas indicada, las cinco primeras y la octava corresponden a un dibujo de un solo paso (entendido un paso como una secuencia *pressed-dragged-released*), la sexta de dos pasos, la séptima de tres pasos, y la novena de tantos pasos como lados tenga el polígono (en este caso, se usará el doble clic para indicar el último vértice). Esto implica que, según la forma, habrá que llevar un control del “paso” por el que se va (y la tarea a hacer en dicho paso). En el caso de las formas usadas en la práctica 7, todas eran de un solo paso (se gestionaban con una sola secuencia *pressed-dragged-released*).

- Texto formateado<sup>3</sup>

## • Atributos de dibujo

El usuario podrá elegir los atributos con los que se pintarán las formas. Para este ejercicio se proponen los siguientes atributos (mejorando los ya incluidos en la práctica 7):

- **Color.** El usuario podrá elegir el color del trazo y el de relleno<sup>4</sup>. Deberán aparecer una serie de colores predeterminados y, además, la posibilidad de lanzar un diálogo de selección de colores.
- **Trazo.** Se podrán modificar el grosor y el tipo de discontinuidad del trazo. En este último caso, se podrán dibujar, al menos, líneas continuas o líneas punteadas.
- **Relleno.** El usuario podrá elegir entre tres opciones a la hora de rellenar: no rellenar, rellenar con un color liso o rellenar con un degradado. En el caso del degradado, éste se aplicará utilizando los dos colores (frente y fondo) seleccionados en ese momento (para la dirección del degradado, se ofrecerá al menos dos posibilidades: horizontal y vertical).
- **Alisado de bordes.** El usuario podrá activar/desactivar la mejora en el proceso de renderizado correspondiente al alisado de bordes.
- **Transparencia.** Se podrá establecer un grado de transparencia asociado a la forma (por ejemplo, mediante un deslizador)<sup>5</sup>
- **Fuente del texto** (en caso de que se incluya el texto como forma). Se podrá establecer la fuente, tamaño y estilo del texto.

Cuando se cambie de una ventana interna a otra, los botones de forma y atributos de la barra de herramientas deberán activarse conforme a la forma y atributos de la ventana activa.

## ■ Algunas recomendaciones

1. El hecho de que cada figura tenga sus propios atributos, implica obligatoriamente la **definición de una jerarquía de clases asociadas a las formas y sus atributos**. Por este motivo, se recomienda incluir en la biblioteca desarrollada en la práctica 7, en el paquete *sm.xxx.graficos*, las nuevas clases diseñadas para abordar este ejercicio.
2. Con un buen diseño de clases, el método *paint* del lienzo tendría que tener la siguiente forma:

```
public void paint(Graphics g){
    super.paint(g);
    Graphics2D g2d = (Graphics2D)g;

    for( XXXX s: vectorFiguras) { //Para cada figura del vector
        // Llamada al método que pinta la forma 's'
    }
}
```

---

<sup>3</sup> La escritura del texto se podrá hacer directamente sobre el área de dibujo o bien utilizando un campo de texto o un diálogo previo en el que introducir la cadena. Independientemente de la forma de introducir el texto, éste deberá de aparecer en el punto de la imagen donde se haga el clic y con el formato indicado.

<sup>4</sup> Nótese que en la práctica 7 no se distinguía entre color de trazo y color de relleno (se usaba el mismo para ambos); en este ejercicio se propone incorporar esta distinción.

<sup>5</sup> Nótese que no es sólo la semitransparencia (como en la práctica 7), sino que el usuario podrá definir el grado de transparencia que vaya desde opaco hasta totalmente transparente.

donde "*s*" será una figura de la jerarquía propia definida en el punto anterior (y "*XXXX*" su superclase). Nótese que en el cuerpo del bucle sólo debe haber una línea de código correspondiente a la llamada al método (externo a la clase del lienzo) que pinta la forma. El único atributo que podría activarse en el código anterior sería "*g2d.clip(areaClip)*" para definir el área de dibujo del lienzo (y, si se desea, la llamada a un método que dibujase un marco alrededor de la imagen), si bien el resto de acciones propias de dibujo deben quedar relegadas a la forma.

3. Las propiedades que puedan estar asociadas al lienzo (color, grosor, etc.) se usarán **para crear las figuras, nunca para pintarlas**. Esto queda claro en el punto anterior, donde en el método *paint* no se activa ningún atributo relativo a las formas (esta tarea queda relegada a cada una de las formas).